Aalto University
School of Science
Degree in Engineering Physics and Mathematics

Tuomo Paavilainen

OPTIMAL GENERATION OF BENT PLATES IN A SOFTWARE PRODUCT

Master's thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Technology.

The document can be stored and made available to the public on the open internet pages of Aalto University. All other rights are reserved.

Espoo, November 19, 2014
Supervisor: Professor Ahti Salo
Instructor: M. Sc. (Phil.) Marko Aho

AALTO UNIVERSITY
SCHOOL OF SCIENCE

ABSTRACT OF THE
MASTER'S THESIS

Author: Tuomo Paavilainen

Title: Optimal Generation of Bent Plates in a Software Product

Degree Program: Engineering Physics and Mathematics
Major subject: Systems and Operations Research
Minor subject: Strategic Management

Supervisor: Prof. Ahti Salo
Instructor: M. Sc. (Phil.) Marko Aho

Due to advances in information technology, Computer Aided Design (CAD) and Building Information Modeling (BIM), designers and architects are nowadays willing to design more complicated steel structures than before, which often containing geometrically complex solids. A support for bent and curved metal plates is a prerequisite for a BIM software to be able to efficiently create such structures.

The goal of this thesis is to study how steel solids that can be manufactured by bending thin metal plates can be modeled in a BIM software product. Also, a modeling approach is presented in which a model of the steel solid can be created based on flat polygon-shaped metal plates in 3-space. Utilizing the model introduced, a decision making problem formulated as a Mixed Integer Linear Programming (MILP) problem is created.

The purpose of the decision model is to automatically decide the parameters and topology of a bent steel solid, when the location and shape of its unbent flat areas are known, and therefore to reduce repetitive work done by an engineer designing steel structures including bent steel plates. A prototype computer program is created of the bent plate data model, the MILP problem and the workflow introduced in the thesis.

The results from the prototype suggest that MILP can in fact be used to implement a supportive algorithm for creating data models of bent steel plate structures in a BIM software product.

Date: 19 November 2014	Language: English	Number of pages: 81

Keywords: Bent steel plates, Steel solids, Unfoldability, Developable surfaces, Mixed Integer Linear Programming

AALTO-YLIOPISTO
PERUSTIETEIDEN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Tuomo Paavilainen

Työn nimi: Optimaalinen taitettujen teräsrakenteiden tuottaminen ohjelmistotuotteessa

Tutkinto-ohjelma: Teknillinen fysiikka ja matematiikka

Pääaine: Systeemi- ja operaatiotutkimus

Sivuaine: Strateginen johtaminen

Vastuuopettaja: Professori Ahti Salo

Ohjaaja: FM Marko Aho

Informaatioteknologian, tietokoneavusteisen suunnittelun ja rakennustietomallintamisen nopean kehityksen ansiosta suunnittelijat ja arkkitehdit haluavat suunnittella entistä monimutkaisempia teräsrakenteita, jotka koostuvat usein geometrisesti entistä haastavammista kappaleista. Rakennustietomalliohjelmiston tulee tukea taivutettuja ja kaarevia teräslevyjä, jotta se olisi kykenevä mallintamaan tällaisia teräsrakenteita.

Työn tavoite on tutkia, kuinka tasaisesta metallilevystä taittamalla valmistettavia metallikappaleita voi mallintaa rakennustietomalliohjelmistossa. Lisäksi työssä esitetään mallinnustapa, jossa tällainen metallikappale voidaan luoda kolmiulotteisessa avaruudessa oleviin tasaisiin monikulmion muotoisiin metallilevyihin pohjautuen. Mallin avulla luodaan lineaarisena sekalukuoptimointitehtävänä formuloitu päätösmalli.

Päätösmallin tarkoituksena on päättää automaattisesti taitetun teräskappaleen parametrit ja topologia, kun teräskappaleen tasaisten, taittamattomien alueiden sijainti ja muoto on tiedossa ja siten vähentää taitettuja teräslevyjä sisältävää teräsrakennetta suunnittelevan insinöörin tekemää toistuvaa työtä. Taitettujen levyjen tietomallista, lineaarisesta sekalukuoptimointitehtävästä ja opinnäytetyössä esitetystä työnkulusta on työssä kehitetty prototyyppitietokoneohjelma.

Prototyypistä saadut tulokset osoittavat, että lineaarisen sekalukuoptimoinnin pohjalta voidaan rakennustietomalliohjelmistoon implementoida algoritmi, joka tukee taitettujen teräslevyrakenteiden tietomallin luomista.

Päivämäärä: 19.11.2014	Kieli: Englanti	Sivumäärä: 81

Avainsanat: Taitetut teräslevyt, teräskappaleet, levitettävyys, tasaantuvat pinnat, lineaarinen sekalukuoptimointi

PREFACE

Completing this thesis finalizes my studies in Aalto University and represents the attainment of my primary goal in life ever since my first day of school almost 20 years ago – obtaining a master's degree. This thesis was done in Tekla Oy as an assignment.

First of all, I would like to thank Tekla Oy for giving me this extraordinary chance to write my master's thesis on such an interesting and challenging topic, and all the support the company has given me. Especially, I want to thank Marko Aho for his guidance and being my instructor. I also want to express my gratitude to Eric Beyer and Petri Hiilinen for providing me with a great deal of knowledge about construction industry needs.

My supervisor Professor Ahti Salo has helped me tremendously in completing my studies. I would like to thank him for all his insight and guiding me through my studies.

I would also like to thank my family, especially my parents who have done their best to motivate and help me to achieve this goal. I would also like to thank my girlfriend Tanja. You made it all so much more bearable.

Tuomo Paavilainen

Table of Content

SYMBOLS

$\widehat{\ }$	Normalization of a vector
^{T}	Transpose of a vector
*	Value of a variable optimizing a function.
$\bar{\ }$	A vector representing a point in 3-space expressed in a coordinate system whose origin is at the singularity point of a cone
A	Area
$\boldsymbol{c}(u)$	Directrix curve of a ruled surface
C_1 and C_2	Constants for calculating a geodesic of a cone
$\boldsymbol{d}$ or $\boldsymbol{d}_{int}$	Direction vector of an intersection line between two planes
$\boldsymbol{e}(u)$	Generator line of a ruled surface
$\boldsymbol{g}(u)$	Geodesic of a surface
h	Thickness of a metal plate or shell or height
$\boldsymbol{h}_0$	The longitudinal axis direction vector in a cylindrical coordinate system
$J(u)$	Length of a directrix curve section from 0 to u
K	Location of the neutral surface relative to thickness
$\boldsymbol{l}_i$	A vector on a polygon edge pointing inside
$L(x_0, x_1)$	Line segment between points x_0 and x_1
L	Length (of a geodesic or a bend)
L_{total}	Total length of a bend including plate extensions
L_i	Bend end segment on side $i \in \{1,2\}$
L_{curve}	Length of the curved section of a bend
m	The steepness of a cone defined as height / radius
M	Moment acting at the end of a metal sheet
$\boldsymbol{n}$	Normal vector of a surface
$\boldsymbol{n}_L$	Normal vector of a line segment
$\boldsymbol{P}_e$	Counter-clockwise ordered polygon on a plane in 3-space
$\boldsymbol{r}_0$	A known polar axis direction vector in a cylindrical coordinate system
R	The radius in a cylindrical coordinate system
$R_{\max}(\theta)$ and $R_{\min}(\theta)$	Radial coordinate value of the outer and inner lateral boundary respectively
$\boldsymbol{s}(u)$	Singular curve of a developable surface

$\boldsymbol{s}_{gd}$	Reference geodesic curve
t	Independent variable of a curve or a line
u	Variable of the directrix curve trajectory function of a ruled surface
v	Generator line independent variable of a ruled surface
v_s	Generator line independent variable value at singularity point
w	Width
x_k^i	The x-coordinate of the kth polygon point on the polygon on side i expressed in the local two dimensional coordinates of the polygon plane.
$x_{\max}(\theta)$ and $x_{\min}(\theta)$	The x-coordinate value of the lateral boundary of a conical bend
$\boldsymbol{x}_{Li}$	A point on end section i
$\boldsymbol{x}$	Location of a point in 3-space
$\boldsymbol{x}_s$	Singularity point of a developable surface, i.e. apex of a cone
$\boldsymbol{x}_{int}$	A point on the intersection line between two planes
x_{2D}	The x-coordinate of a point on an unfolded plane
y_k^i	The y-coordinate of the kth polygon point on the polygon on side i expressed in the local two dimensional coordinates of the polygon plane.
$x_{\max}(\theta)$ and $x_{\min}(\theta)$	The y-coordinate value of the lateral boundary of a conical bend
y_{2D}	The y-coordinate of a point on an unfolded plane
z	Level of the neutral surface
α	Angle between planes. Defined as the angle between the normal vectors of the planes.
β	Slope of a line (discontinuity boundary) on a plane defined by a polygon
β_0	A constant replacing the variable β in the linear approximation of bend length
γ	The independent variable of a line between two points on the unfolded plane
δ	The independent variable of a line between two points on the unfolded plane
$\boldsymbol{\delta}_i$	Discontinuity boundary line on side $i \in \{1,2\}$
σ	Stress acting on a fiber in metal
ϵ	Strain of a fiber in metal
ϵ_0	A small real number

$\boldsymbol{\epsilon}$	Array or binary variables indicating the set of selected vertex points, or a vector with a small length.
κ	Curvature
$\lambda(u)$	Function defining the location of the first lateral boundary of a cylindrical bend
ξ	A value defining how much the user is willing to trade off additional bend width for additional length.
Σ	Summation or a set of elemental plate surface points in space
Ψ	Location of the neutral axis/surface in absolute terms
θ	The azimuth angle in a cylindrical coordinate system
θ_0	The azimuth angle of a conical bend at the discontinuity boundary, half of the opening angle of a
Δ_{max} and Δ_{min}	Distance from the reference geodesic to the outer and inner lateral boundary respectively, along the generator line
$\omega(u)$	Function defining the location of the second lateral boundary of a cylindrical bend

1 INTRODUCTION

The purpose of this thesis work is to study how bent steel structures can be modeled in a software product, and based on that knowledge, develop an algorithm that helps the software user to easily create the most common manufacturable bent plate structures. The basic concepts and research problems are presented as well as some of the most relevant literature is reviewed. In the thesis, a bent plate modeling concept shall be sketched and a prototype be built. Similar concepts have been developed for instance by Yeh et al. (1995). Linear programming has not been widely used in the context of CAD or BIM, but Yang & Chuang (1994) have adopted a somewhat similar approach for solving a relatively similar problem, yet from a different perspective and optimizing a different objectives.

Tekla Structures is a Software Product for designing and detailing steel structures. It is the main tool of many structural engineers working for construction firms, engineering offices and steel fabrication companies. Its current version 20.1, however, does not support bent steel plates.

Bending and folding of plates is a relevant subject of study in many fields. Dating back to the old Japanese tradition of paper-folding or origami, folded and bent plates are used for example in package design, ship design, construction and architecture, automobile design, and other industrial design. (Liu & Tai, 2007; Yeh et al., 1995; Pottmann et al., 2008)

In sheet metal fabrication, the work starts with a flat piece of sheet metal that is thereafter bent, curved and otherwise machined. However, when designing a bent plate structure with a CAD tool or a BIM tool like Tekla Structures, the designer first designs a three-dimensional description of the metal part, based on which the computer program should be able to compute sufficient directions for the fabricator (either a person or a machine) to be able to produce the part. Chuang & Huang (1996) have studied a similar problem.

Due to technological advances and increased use of BIM, there is an increasing demand for more complex steel designs, often containing bent steel plates. Also designers are accepting assignments with more complexity than before. To tackle the complexity and to be able to model bent plates in an effective manner, an easy-to-use tool is needed for

modeling the bent plates, unfolding the bent plate into the unbent state and creating the drawings and automatic machining instructions (computer numeric control) necessary for fabrication.

Eventually architects are hesitant to hold to conventional plane surfaces and modern architecture utilizes curved surfaces. More and more architectural projects also include double curved structures and so called free-formed structures where they do not follow a strict form. It is therefore obvious, that modern engineering and steel detailing tools ought to support curved and possibly double curved structures. (Pottmann et al., 2008; Eigensatz et al., 2010)

The introduction of a bent plate concept to a BIM software like Tekla Structures would affect many stakeholders in the process of designing and building bent plates structures. Architects and designers do not want to be tied to simple, straight structures, engineers often face situations where they need to model bent shapes. Steel detailers also need bent steel plates in their work, and they need to provide the production information to the fabricators, who eventually manufacture the bent shapes from raw material.

The modeling of the bent plate structures would have to be done in an intuitive manner suiting the most common use cases. In addition to the restrictions introduced by the workflow of the end user, physical and geometrical minimum requirements have to be satisfied. This thesis aims at finding a simple modeling method for a civil engineer to model feasible bent steel plate structures so that the user needs to provide a minimal amount of input.

The thesis is structured as follows. Section 2 reviews the literature concerning topics related to bent steel plates and the software presentation of solid objects. Thereafter, in Section 3 the concepts and terminology used in the thesis are presented. In Section 4, the methods for generating and verifying the feasibility of the software presentation of bent plates are introduced. In Section 5, results are given and finally Section 6 draws conclusions.

2 LITERATURE REVIEW

2.1 MATHEMATICAL MODELING OF BENT PLATES AND SHELLS

Bending of metal plates is widely discussed in literature. One can get a good understanding of the classical theory of plates and shells from literature. (Timoshenko et al., 1959; Timoshenko & Gere, 2012; Reddy, 2006; Tongxi & Zhang, 1996).

In this thesis, classical theory of plates and shells is applied to plates of uniform thickness and a single isotropic material. Because we are defining a concept for permanently deformed (bent) plate structures, plastic deformations of metals are to be studied.

Kinematics is the part of classical mechanics that studies the motion of a point or a set of points in material without taking into account the forces causing the movement. "Kinematics is the study of the geometry of motion" (Beggs, 1983). The kinematic study of deformation of a body is mainly concerned with the geometry of the displacement of the body. In the thesis, we aim to understand the geometry of the bent plates – thus kinematic properties of plane bending are studied.

There are two distinct theories of plastic bending, the *engineering theory of plastic bending* and the *mathematical theory of plastic bending*. The engineering theory of plastic bending was developed mostly in the beginning of the 20th century and is relatively simple but enough for many engineering applications. The engineering theory of plastic bending assumes that the centroidal axis, the neutral axis and the unelongated axis are all in the same place. It also assumes that the cross sectional shape of the body will not change during bending, only considers one-dimensional stress and neglects the traverse sheer stress on cross sections (Tongxi & Zhang, 1996).

The neutral surface is the surface inside the metal where the fibers do not experience stress or strain. During bending, on the outer side of the neutral surface, the fibers of the metal are under tension and on the inner side of the metal they get compressed. Neutral axis is the cross section of the neutral surface and the cross section of the bent object perpendicular to the angle of the bent section. In two-dimensional cases, where the bends are cylindrical, it is enough to study the neutral axis, because it is the same with all cross sections of interest. This is not the case with non-cylindrical bends such as bends with a

conical geometry. In Figure 1, the displacement of the neutral axis due to plastic bending is depicted.

The unelongated layer is the surface of the bent section whose area remains the same before and after bending. An unenlongated axis is a similar two-dimensional concept for the unenlongated layer as the neutral axis is for the neutral surface. Also, all the unenlongated axes have the same length before and after bending.

The central axis of the bend is in the geometrical center of the cross section of the plate to be bent.

2.2 BENDING OF A SIMPLE BEAM

In the *engineering theory of plastic bending*, one assumes that the neutral axis, the unenlongated axis and the central axis coincide. In the simplest case, let us consider a rectangular long beam that consists of fibers parallel to the length of the beam. The beam is subject to two moments $M_1 = M$ and $M_2 = M$ acting on its ends, bending the whole beam creating a cylindrical bend. In the beginning, the neutral axis lies at the center of the beam. When starting to bend, all the fibers on the convex side of the neutral line experience tension and on the concave side, compression. The stress-strain relationship for the fibers is different for compression and tension:

$$\sigma = \begin{cases} f_t(\varepsilon), \varepsilon \geq 0 \\ f_c(\varepsilon), \varepsilon \leq 0' \end{cases} \quad (1)$$

where σ denotes stress, and ε denotes strain. Because of the round shaped intersection of the bend and the fact that the neutral fiber experiences no strain (or stress) if the unenlongated and the neutral fiber coincide, the strain for a given fiber is proportional to the distance of the fiber from the neutral axis z:

$$\sigma(z) = \begin{cases} f_t(z\kappa), z \geq 0 \\ f_c(z\kappa), z \leq 0' \end{cases} \quad (2)$$

where κ is the curvature of the bend.

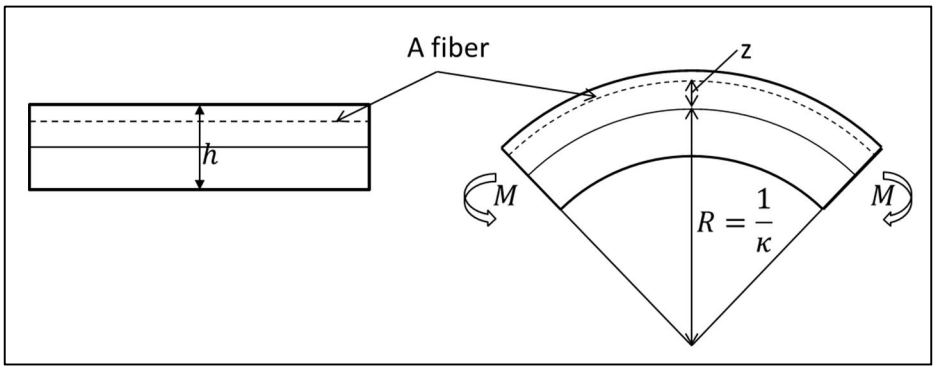


Figure 1. Bending of a simple beam using the engineering theory of plastic bending

According to Nádai (1950), the following conditions need to apply to pure bending in the classical engineering theory of plastic bending:

$$\begin{cases} \iint_A \sigma dA = 0 \\ \iint_A \sigma z dA = M, \end{cases} \tag{3}$$

where A is the cross section around the longitudinal axis of the beam. Because of the rectangular shape of the beam intersection around its longitudinal axis, and because of the formulation of the stress strain relationship in (2) and (3) can be written as

$$\begin{cases} \int_{z_1-h}^{z_1} \sigma(z) dz = 0 \\ \int_{z_1-h}^{z_2} \sigma(z) z dz = \frac{M}{w}, \end{cases} \tag{4}$$

where h is the height of the beam and w the width. We know that the neutral axis lies on the level $z = 0$, and that the top surface of the bean on the level $z = z_1$. But here the value of z_1 is unknown. Thus (4) can be used to evaluate the location of the neutral axis. Because the stress σ is of a different form for fibers on the inner side of the bend (below

the neutral axis) and on the outer side (above the neutral axis) of the bend, and because z_1 is not known, the location of the neutral axis is a function of the curvature of the bend

$$z_1 = \Psi(\kappa). \quad (5)$$

2.3 THE SOFTWARE PRESENTATION OF SOLID OBJECTS

The solid representation models aim to determine whether a given point in space is within a solid or not. "The objective of solid modeling is to represent, manipulate, and reason about; the three-dimensional shape of solid physical objects, by computer" (Hoffmann & Rossignac, 1996). Adequate and suitable representation technique is needed to efficiently perform the latter stages of solid modeling: manipulation and reasoning. In this thesis, modeling of only physically feasible solids of single homogenous material are considered.

Several techniques for representing solid objects in software tools have been developed and are used. However, the most used approaches for solid representation are the following (Hoffmann & Rossignac, 1996; Stroud, 2006):

- Spatial subdivision
- Constructive solid geometry (CSG)
- Boundary representation (Brep)

In CSG, a solid object is represented as a composite of primitive solid objects using set theoretic Boolean expressions. The primitive solid objects are traditionally block, sphere, cylinder, cone and torus. The Boolean operations used are normally union, intersection and difference (Voelcker & Requicha, 1977; Hoffmann & Rossignac, 1996). Thus, for example a capsule can be represented using two spheres and a cylinder.

Spatial subdivision methods decompose solids into simple geometrical objects, so called cells. Spatial subdivision methods are further divided into two groups: *boundary conforming* and *boundary approximating* representations. Boundary conforming representations are such that the space is divided into geometries that exactly define boundaries of the solid. An example of such a representation is a binary space partition tree (BSP tree). BSP tree is a binary tree structure whose nodes are separator planes that divide the space into two different subspaces. Those subspaces can either be further

divided into two subspaces (a sub-tree) or be considered as the leaves of the tree structure. The leaves of the BSP tree represent the points that either belong or do not belong to the solid, depending on the value of the leaf.

Boundary approximating spatial subdivision methods divide the space into a standard division such as a grid. Each cell either belongs or does not belong to the solid. Thus, such a representation can only approximate objects that are not directed along the grid. (Hoffmann & Rossignac, 1996; Stroud, 2006)

In a boundary representation model (Brep), the object solid is modeled in terms of its boundaries. A Brep model consists of topological objects (faces, edges and vertices) and geometric information of the topological objects. The topological objects define the topological structure of the model, that is, whether an object consists of the other, or whether they are adjacent. The topological and geometric object together define which of the points of the space belong to the object, which do not.

2.3.1 THE BASIC BREP DATA STRUCTURE

In a Brep model, the topological data and geometrical data are separated. The topological units: vertices, edges and faces define the dependencies and topology or structure of the "skin" of the solid, whereas the equivalent geometric objects: points, curves and surfaces define the shape and location of the equivalent geometry.

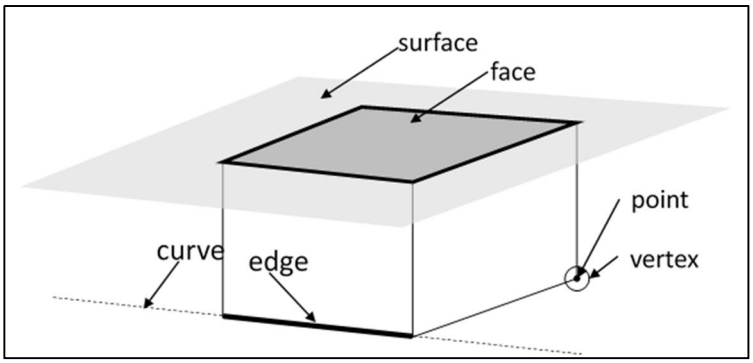


Figure 2. Surface, curve and point are the geometric equivalents of the topological objects face, curve and vertex, respectively.

The data structure of a Brep is natural to be grouped into two categories, the topological and geometrical variables. A simple Brep data structure is as follows based on the modeling principles in Mäntylä (1988):

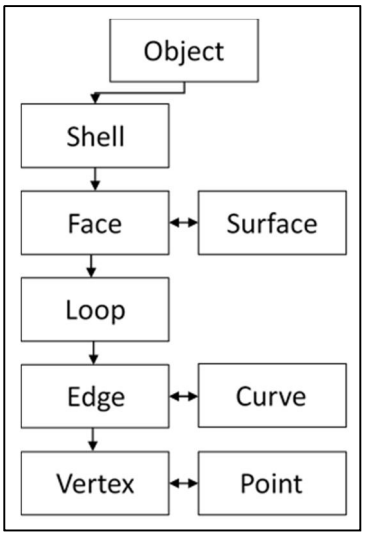


Figure 3. A simple Brep solid data structure

On the left column of the Figure 3, the topological types are listed. On the right side the associated geometrical objects are shown. Two new types, shell and loop, are also introduced. The topological types in a BREP solid presentation are defined as follows:

> *VERTEX: A vertex is a node of the datastructure, lying at a point in space*
> *EDGE: An edge is a segment of a curve, running between two vertices. …*
> *FACE: Faces are portions of surfaces. Faces are bounded by loops, which are ordered sets of edges.*
> *LOOP: In its simplest form, the model datastructure consists only of faces, edges and vertices. However, this does not allow multiply connected faces, where there is an outer boundary and one or more inner boundaries. To allow this kind of model the edges bounding a face are divided into closed circuits of edges, called Loops.*
> *…*
> *SHELL: Each closed set of faces in the object forms a Shell. It is useful to represent these shells explicitly in some way in a model, rather than having to retrieve the information by traversing a faceset to see if it is closed.*
> *…*
> *POINT: A zero-dimensional entity, a position in 3D Euclidean space defining the position of a vertex.*
> *CURVE: A one-dimensional entity defining the shape of an edge.*
> *SURFACE: A two-dimensional entity defining the shape of a face …*
> *(Stroud, 2006)*

2.4 GEOMETRY REQUIREMENTS

In this section the requirements for the geometry of an unfoldable surface are discussed. Unfoldability is discussed extensively in literature, and a surface needs to satisfy four different prerequisites so that it can be unfolded onto a plane: developability, local geometrical unfoldablility, global geometrical unfoldability and topological unfoldability. All four are discussed in this section.

2.4.1 DEVELOPABILITY

First, a plate with a zero height is considered, so that only the bending of surfaces can be studied. According to Pottmann et al. (2008) a surface can be unfolded onto a plane so that the distances remain the same before and after unfolding, if the surface is *developable*. All developable surfaces in 3-space are ruled surfaces, so developable surfaces can be parameterized using the parameterization of a ruled surface (Peternell, 2004):

$$\boldsymbol{x}(u,v) = \boldsymbol{c}(u) + v\boldsymbol{e}(u), \tag{6}$$

where $\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{e} \in \mathbb{R}^3$ and $u, v \in \mathbb{R}$. The function $\boldsymbol{c}$ is called the directrix curve. A ruled surface is such, that it can be built of a set of lines. These lines – *the generator lines* – can be found by fixing the value of u in (6). The surface is developable, if all the points on a generator line have the same tangent plane, i.e. they all have the same normal vector direction. The normal vector of a ruled surface can be expressed as:

$$\boldsymbol{n}(u,v) = \boldsymbol{c}'^{(u)} \times \boldsymbol{e}(u) + v\boldsymbol{e}'^{(u)} \times \boldsymbol{e}(u)\,. \tag{7}$$

If the normal vectors are the same for all $v \in \mathbb{R}$, when u is fixed, the direction of the normal vector has to be the same independent of the parameter v. This is the case when $\boldsymbol{c}'(u) \times \boldsymbol{e}(u) \parallel v\boldsymbol{e}'(u) \times \boldsymbol{e}(u)$, which is equivalent to

$$\boldsymbol{c}'(u) \times \boldsymbol{e}(u) \parallel \boldsymbol{c}'(u) \times \boldsymbol{e}(u). \tag{8}$$

This condition is equivalent to the vectors $\boldsymbol{c}'(u)$, $\boldsymbol{e}(u)$ and $\boldsymbol{e}'(u)$ being on a plane, thus being linearly dependent

$$\det(\boldsymbol{c}'(u), \boldsymbol{e}(\boldsymbol{u}), \boldsymbol{e}'(u)) = 0. \tag{9}$$

A developable surface can have a singularity point, where the generator lines meet. At such a point, the normal vectors of the curve are not well-defined. When approaching the singularity point, the value of $\boldsymbol{e}$ approaches zero, and thus the proportional length of the normal vector approaches the length of the first term of the expression in (7):

$$\frac{(\boldsymbol{c}'(u) \times \boldsymbol{e}(u) + v_s \boldsymbol{e}'(u) \times \boldsymbol{e}(u))^2}{(\boldsymbol{c}'(u) \times \boldsymbol{e}(u))^2} = 1, \tag{10}$$

where v_s is the value of the variable v at the singularity point. From the equation above, we get by substitution

$$v_s = -\frac{\big(\boldsymbol{c}'(u) \times \boldsymbol{e}(u)\big) \cdot \big(\boldsymbol{e}'(u) \times \boldsymbol{e}(u)\big)}{\big(\boldsymbol{e}'(u) \times \boldsymbol{e}(u)\big)^2}. \tag{11}$$

The singularity point defines the type of the developable surface. If there is one single singularity point at infinity, the surface is cylindrical. If there is one single finite singularity point, the surface is conical. In the third case, the singularity points form a space-curve, the singular curve $\boldsymbol{s}(u) = \boldsymbol{x}\big(u, v_s(u)\big)$. (Peternell, 2004)

2.4.2 EXCLUSION OF OVERLAPPING UNFOLDED SURFACES

Because the shells must be unfolded onto a single metal plate, i.e. the three dimensional object has to be able to be made out of the two dimensional (unfolded) object, the folded object must have no overlapping surfaces in the unfolded object. In Figure 4, there is an example of a legal and an illegal unfolded object. The property of no overlapping surfaces arising is called *global geometric unfoldability* (Wang, 1997).

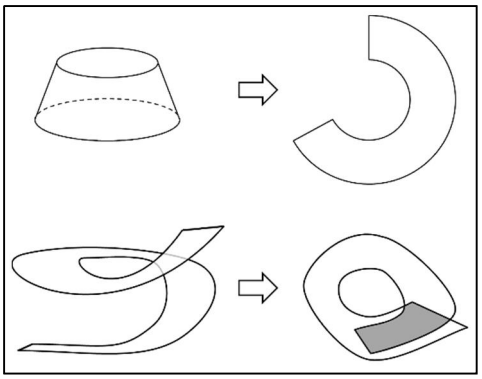


Figure 4. Unfolding a legal object and an overlapping object.

The validation of the unfolded surface for overlapping polygons is generally solved by first unfolding the part and after that verifying that the two dimensional shape does not overlap itself, see for instance (Liu & Tai, 2007; Tai et al., 2004). In Liu & Tai (2007),

the problem has been solved for a polygon. There are many algorithms for finding whether a simple polygon is self-intersecting or not. The detection of self-intersection in the case of a general closed loop curve is discussed by Pekerman et al. (2008).

2.4.3 LOCAL GEOMETRIC UNFOLDABILITY

Local geometric unfoldability is defined in terms of *corners*. "*A corner is defined as the intersection of three or more connecting faces.*" (Wang, 1997) A corner is locally unfoldable, if the sum of the face angles at the corner exceeds 2π. The face angle is defined in "unfolded" terms, meaning that for instance in the case of unfolding a sector of cone surface through the singularity point (apex) of the cone, and the singularity point coincides with the *corner* under investigation, the face angle is the angle of the corner on the unfolded flat surface. If an object is globally geometrically unfoldable, it is also locally geometrically unfoldable. Therefore, it is not always necessary to verify the local geometric unfoldability.

2.4.4 TOPOLOGICAL UNFOLDABILITY

Topologically unfoldable part is such that all the connections of the part remain connected when unfolding occurs. Intuitively it means, that the part should be such that it can be unfolded without cutting. Topological unfoldability has to be verified during the process of the user modeling the part.

Yeh et al. (1995) suggest an algorithm based on face-edge-graph to verify the topological unfoldability of a sheet-metal part. In a face-edge graph, the faces of the (zero-thickness) plate composite are represented by the nodes of the graph. The connecting edges that connect the faces together with bend sections are represented by the linkages of the graph.

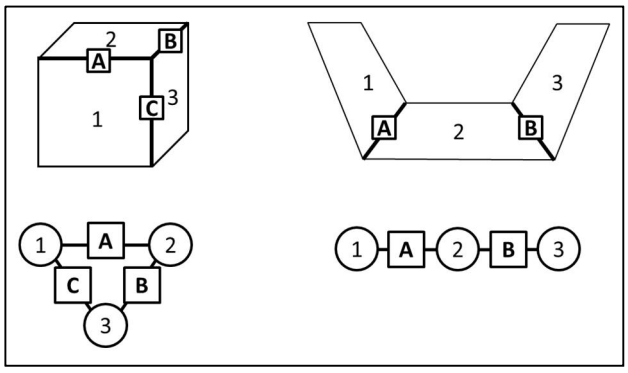


Figure 5. Face-edge -graph presentation of two sheet metal parts.

If the graph can be presented as a tree, the part is always topologically unfoldable. A graph being a tree is equivalent to

$$n - l = 1, \tag{12}$$

where n is the number of faces and l is the number of edges in the graph.

In the case of a tree (or a chain), the part can easily be unfolded. In the case of a loop, unfolding cannot necessarily be done. This is illustrated in Figure 5. Even though, it is enough for the edge-face to be a tree in order to be unfoldable, there are cases where the graph is not a tree, but still it can be considered topologically unfoldable. Such an example is given in Figure 6.

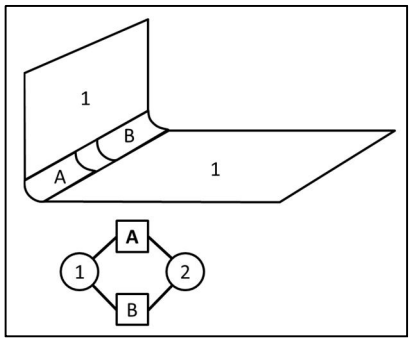


Figure 6. A bendable object with a loop in its face-edge graph

Such cases with loops in the face-edge graph that can still be unfolded, can however be reduced into the form of a tree formed face-edge graph. The algorithm presented in Yeh et al. (1995) groups the nodes in sub graphs, and determines whether the group is topologically unfoldable. If the group is topologically unfoldable, it also checks whether the folded surfaces have “concentric arcs”. This is essentially a special case of verifying that the folded surfaces (for example A and B in Figure 6) are subspaces of the same developable surface.

3 CONCEPTS AND TERMINOLOGY

In this section the concepts and terminology used in the thesis are introduced. Because the aim of this thesis is to introduce an automated way of creating a model of a metal shell in a computer program, the concepts on which the model is based are presented here.

3.1 BENT PLATE STRUCTURES

A bent plate structure is a data model or a software presentation that corresponds to a steel shell that can be produced by bending a flat steel plate possibly several times. In the thesis, a bent plate structure can have areas that are planar and areas that are curved, as long as it can be fabricated from a flat steel plate. The result is an object in 3-space that consists of straight, unbent plates and bends that connect the unbent plates. The simplest case is a plate with one *parallel bend* with a zero-radius.

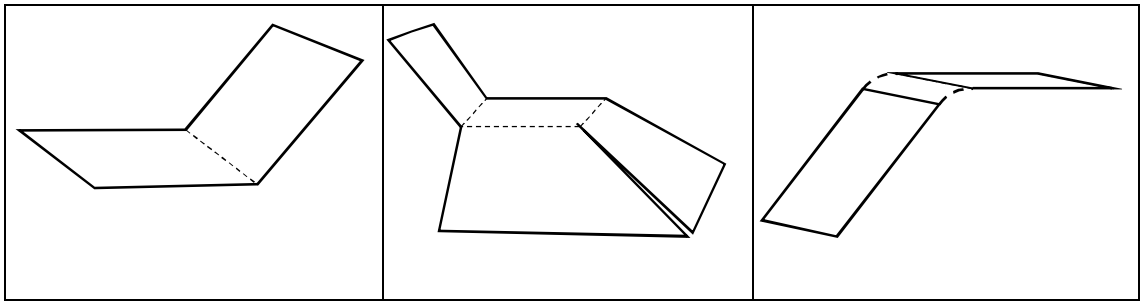


Figure 7. A bent plate with a single parallel bend with a zero radius.

The object depicted in figure Figure 7.a consists of two unbent parts and one bend. A bent plate can have several bends as in Figure 7.b. Also, it is often not possible to assume that the bends are totally sharp, i.e. have a zero radius. A bend with a positive radius is shown in Figure 7.c.

In all of the example bent plate structures mentioned above, the two edges connected with a bend are *parallel bends*. This means that all the folding lines are parallel to each other. A folding line is a line on a flat plate around which the plate is then folded. A bend with a cylindrical shape can be thought of consisting of an infinite number of folding lines, along which the plate is bent. *Non-parallel bends* are also studied in the thesis, i.e. bends

whose connected edges and folding lines are not parallel to each other. A conical bend presented in Figure 8 is an example of a non-parallel bend.

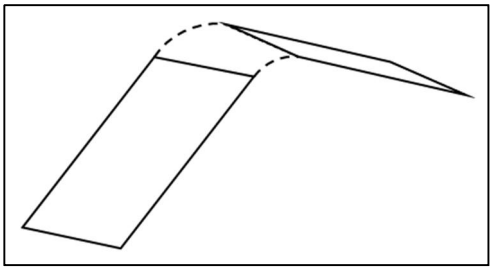


Figure 8. A bent plate with a single non-parallel bend.

Often steel plates are bent in order to create pipes or structures of a form of a section of a cone. Unlike all the previous examples, a pipe is not a connecting bend, and only one edge (and a radius or radii and the direction) is needed to define this structure.

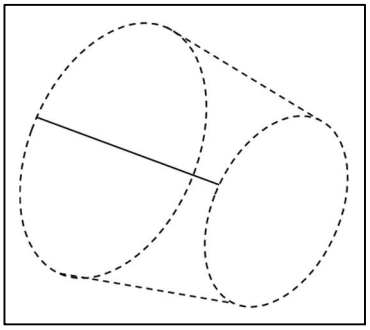


Figure 9. A truncated cone.

3.1.1 BENT PLATE STRUCTURE TOPOLOGY SOFTWARE PRESENTATION

In order to model a bent plate structure as described above, a graph structure is needed with both bends and non-bent plates to describe the connections (topology) of the bent plate structures. Based on the description above, both the folded (3D model) and unfolded version (2D shape) of the bent plate structure has to be able to be generated. A tree is a graph structure that always produces a topologically unfoldable structure. For that reason, and because this thesis does not focus on topological unfoldability, a bent plate structure is modeled as a tree of straight and bent sections in the context of this thesis. An example of a bent plate topology software representation is presented in Figure 10.

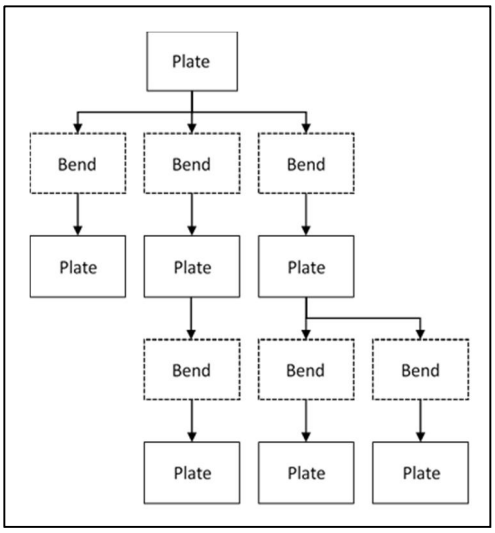


Figure 10. Bent plate tree structure

3.2 ELEMENTAL PLATES

Elemental plates are geometrical objects that correspond to flat areas of a steel shell made of a flat steel plate by bending. A bent plate structure consists of (also) elemental plates. In this section, elemental plates are defined as sets of points in space belonging to the flat area of the steel shell.

The concept of an elemental plate is introduced because of the chosen modeling approach, where a bent plate is modeled by introducing the geometry and locations of flat steel plates as they are located as a result of bending. The reasoning behind this approach is, that the flat areas' orientation and location may be connected to other objects by a welding or a bolt. The approach is thus chosen to make such connections (typical in BIM models) possible.

Elemental plate surface:

The elemental plate surface Σ is all the set of all the inner points of a simple polygon P_e. For example, ray intersection method (Huang & Shih, 1997) can be used to find out whether a point is inside a polygon.

$$\boldsymbol{P}_e = \begin{bmatrix} \boldsymbol{x_1} \\ \vdots \\ \boldsymbol{x_n} \end{bmatrix}, \tag{13}$$

where $\boldsymbol{x}_i \in \mathbb{R}^3$, and n is the number of the polygon points.

The polygon points x_i are on a plane. Let the vector normal to the polygon edge from points $\boldsymbol{x}_i$ to $\boldsymbol{x}_{i+1}$ be denoted $\boldsymbol{l}_i$. This vector is also on the same plane as the points and it is pointing inside the vector from the edge.

Elemental plate: An elemental plate is the set of points **y**, for which **y** is less or equal than the distance $\frac{h}{2}, h \in \mathbb{R}$ from an elemental plate surface point $\boldsymbol{x}$, and the distance vector is perpendicular to the elemental plate surface.

Elemental plate border face: An elemental plate border face is the points on the edge of the elemental plate. The top and the bottom of the elemental plate are not plate border faces. An elemental plate border face E_i:

$$\begin{aligned} \boldsymbol{x} \in E_i, \text{iff } \exists (a_i, b_i), \boldsymbol{x} &= \boldsymbol{x}_i + a_i h \boldsymbol{n} + b_i(\boldsymbol{x}_j - \boldsymbol{x}_i) \\ -\tfrac{1}{2} &\leq a_i \leq \tfrac{1}{2} \\ 0 &\leq b_i \leq 1 \\ j = 1, \text{if } i = n, &\text{otherwise } j = i + 1, 1 \leq i \leq n, i, j \in \mathbb{Z}^+. \end{aligned} \tag{14}$$

The elemental plate edge normal points out of the elemental plate. I.e. it is the same direction as $-\boldsymbol{l}_i$.

Elemental plate border segment: The middle section segment and the normal of an elemental plate border face. The middle segment is also the corresponding edge of the Polygon P_e. The elemental plate border segment has an infinite number of normal vectors, but in this thesis when referring to elemental plate border segment the vector $-\boldsymbol{l}_i$ is referred.

3.3 BENDS

A bend is the part of a bent plate structure that is not part of the elemental plates. A bend is here defined as points in space around the neutral surface of the bend.

The K –coefficient: The K –coefficient determines how far away the inner face of the bent shell is from the neutral surface of the shell. This coefficient is based on the metal properties and the local curvature of the bend. (Timoshenko & Gere, 1961)

Bend end segment is a line segment that belongs to and lays at the edge of the bend neutral surface. Bend end segment is a line segment $L(\boldsymbol{x}_0, \boldsymbol{x}_1) = \{\boldsymbol{x} = \boldsymbol{x}_0 + a(\boldsymbol{x}_1 - \boldsymbol{x}_0) | 0 \leq a \leq 1\}$. It also has a normal vector $\boldsymbol{n}_L$ that is perpendicular to the segment L.

Bend neutral surface: Let S be a smooth, developable 2-manifold in $\mathbb{R}^3$. The surface S is defined using the parameterization of a ruled surface so that

$$\boldsymbol{x}(u, v) = \boldsymbol{c}(u) + v\boldsymbol{e}(u), x, e, c \in \mathbb{R}^3, u, v \in \mathbb{R} \Leftrightarrow \boldsymbol{x} \in S. \quad (15)$$

The normal for the bend neutral surface is calculated using (7). Let us denote the normal $\boldsymbol{n}(u, v)$.

If S is a bend neutral surface, it fulfills the following conditions:

1. $\forall \boldsymbol{x}_i \in L_i \, \exists \boldsymbol{y}_i \in S \, s.t. \; \boldsymbol{y}_i - \boldsymbol{x}_i = \left(K(y_i) - \frac{1}{2}\right) \mathrm{N}(u_i, v_i)$
2. $\boldsymbol{n}(\boldsymbol{y}_i) \perp \boldsymbol{n}_{L_i}$
3. $f(\boldsymbol{y}_\epsilon + \boldsymbol{\varepsilon}) = 0 \Rightarrow f(\boldsymbol{y}_\epsilon - \boldsymbol{\varepsilon}) \neq 0$, where $\boldsymbol{y}_\epsilon, \boldsymbol{y}_i \in L$and $\|\boldsymbol{\varepsilon}\| \to 0$ and $\boldsymbol{n}_{L_i} + \boldsymbol{\varepsilon} > \boldsymbol{n}_{L_i} - \boldsymbol{\varepsilon}, i \in \{1, 2\}$.

L_1 and L_2 are bend end segments.

Bend: A bend is defined as a shell with a constant thickness around the bend neutral surface. The relative distance of the shell bottom surface from the shell neutral surface is given by the K –coefficient. The points x belonging to the bend B can be expressed as follows:

$$x \in B, \text{iff} \, \exists y \in S, d \in \mathbb{R} \mid x = y + dh\frac{\nabla(y)}{|\nabla(y)|}, \text{where} \; -K(y, f) \leq d \leq K(y, f) - 1$$

$0 \leq K(y) \leq 1$.

4 BEND FITTING

In this section a method for creating bent plate structures is proposed. Many CAD software tools such as SolidWorks and SpaceClaim Engineer support features for modeling bent plate structures. In these tools the folded and bent plate structures are modeled as 3D-objects independent of their environment. In a BIM tool like Tekla Structures, the connections between 3D-objects play a more important role, and in many cases, the bent plate structure should be generated based on its environment. Consider a case where two steel beams are connected with a folded steel plate bolted to them. If then either of the beams is moved, the folded plate should change accordingly, so that it still connects the beams. This modeling approach was selected as a result of discussions with Tekla engineers.

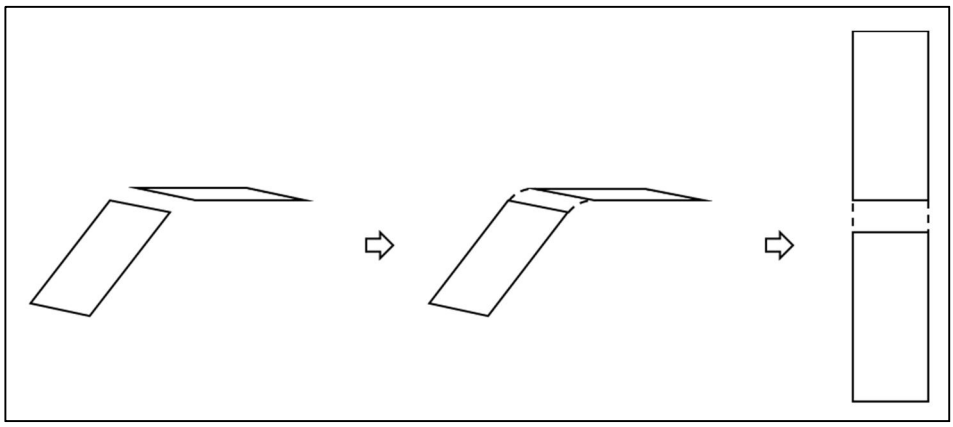


Figure 11. Bent plate modeling approach

For this reason, a method for creating bent plate structures based on connecting straight metal plates is introduced. A bent plate structure is modeled by introducing the end result first and the data model, a 3D model and an unwrapped drawing are automatically generated by the software. The end user models the plates to be connected, specifies how the plates should be connected, and provides the parameters of the bends. The software verifies that the parameters are correct and that the bent plate structure is feasible. Thereafter it automatically fits the bend between the plates and creates the model of the bent plate structures. The user is also given the possibility to edit the bent plate object after the initial specification.

This method consists of connecting plates consists of three partially interconnected tasks:

- Selecting the edges of both planes that are connected,
- Fitting a curved section between the edges,
- Defining the shape of the bend boundaries.

First in this Section 4.1 the fitting of the curved section is discussed. In that part, fitting of two special cases of developable surfaces, a cylindrical surface and a conical surface, between a known set of plate edges is studied.

In Section 4.2, different bend boundaries and their parameterizations are discussed and ways to define the borders of a bend are suggested. Thereafter in Section 4.4, unfolding of a developable surface is discussed, because bend boundaries are often selected so that they fulfill certain requirements that require understanding of how the bent section can be unfolded. Then aspects of selecting the bend boundary parameterization for a bend are discussed and formulas for some selected parameterizations of bend boundaries are derived in Section 4.5. This is done for curved sections following a cylindrical or conical form. Finally in Section 4.6, it is discussed how the edges to be connected can be automatically selected in such a way, that the decision satisfies a considerable portion of common use cases. An algorithm based on Mixed Integer Linear Programming (MILP) is proposed for automatically determining the topology of the bend.

The steel plates and shells in this section are assumed to have a thickness of zero. It is assumed that the plates have a zero thickness, because the focus of this thesis is to understand how elemental plates can be automatically connected to form unfoldable geometries, and the location of the neutral surface is not essential for solving the best way to connect elemental plates with a curved section. If it is needed at a later point, a positive thickness can be assumed, and either the central or the neutral surface and be adjusted to follow the theory of plastic bending as introduced in Section 2.2.

4.1 BEND FITTING PROBLEM

In order to construct bent plate structures, we first need to fit two elemental border faces with equal heights together with a bend. Because the heights of the bends are equal, it is enough that the bend end middle segments are equal to the given elemental plate border segments.

Such a function $f(\boldsymbol{x}), x \in \mathbb{R}^3$ needs to be found that satisfies the requirement conditions of a bend neutral surface with the given end middle sections.

Before we can achieve that, we have to know the form of the functional $K(y, f)$. Because all the shells are of a constant thickness and of the same material, (5) can be used. According to it, the location of the neutral axis is a function of the curvature of the bend. Because K is the relative location of the neutral surface.

$$K = K(\kappa(y)). \tag{16}$$

The bend fitting problem can be expressed as follows:

> Given two elemental plate border segments, find the bend neutral surface, i.e. the bend end segments L_1, L_2 and a developable surface S, so that S fulfills the conditions for a bend neutral surface and L_1 and L_2 are located $\left(K(\kappa) - \frac{1}{2}\right) h$ distance from the corresponding elemental plate border segment, on the elemental border face.

Because the neutral surface is always ruled and developable, and directed according to the normals of the elemental plate border section, the neutral surface is a constant distance from the middle segment of the elemental plate border face. Thus the bend end segment and corresponding elemental plate border segment are a fixed distance from each other and the same direction. Thus it is enough to fit the middle surface first, build the bend geometry based on the middle surface, and only when unfolding the geometry, base the unfolding on the location of the neutral surface. Because of this, and the fact that this thesis focuses on fitting the bends, bent plates are assumed to have a zero thickness, and concentrate on fitting developable surfaces.

The minimum information to define a developable surface is to define its directrix curve and its singularity point or –curve. Let us consider the requirements of the input

(elemental plate border segments) in three different types of developable surfaces, cylindrical surface, conical surface and the case of a general singularity curve.

4.1.1 CYLINDRICAL SURFACES

First the simplest case, the case of a cylinder is considered. In the case of a cylinder, ruled surface parameterization in (15) can be formulated as follows, because the vector $\boldsymbol{e}$ is constant.

$$\boldsymbol{x}(u,v) = \boldsymbol{c}(u) + v\boldsymbol{e}. \tag{17}$$

Now, let us parameterize the end section L_1.

$$\boldsymbol{x}_{L1}(t_1) = \boldsymbol{x}_{0,L1} + t_1\boldsymbol{e}_{L1}. \tag{18}$$

The section is part of the surface. There has to be a mapping that maps the end section parameter w_1 to the parameters of the cylinder u, v.

$$\boldsymbol{x}_{0,L1} + t_1\boldsymbol{e}_{L1} = \boldsymbol{c}(u) + v\boldsymbol{e}. \tag{19}$$

By noting $u = u(t)$ and $v = v(t)$ and solving the equation above for $\boldsymbol{c}(u(t))$ we can see that the values of $\boldsymbol{c}$ have to be on a plane defined by the section L_1 and the vector $\boldsymbol{e}$

$$\boldsymbol{c}(u(t)) = \boldsymbol{x}_{0,L1} + t_1\boldsymbol{e}_{L1} - v(t)\boldsymbol{e}. \tag{20}$$

Thus we get the intuitive result that the surface $\boldsymbol{x}(u,v)$ is either a plane around the end section L_1 or $\boldsymbol{e}_{L1} \parallel \boldsymbol{e}$. Clearly the same result applies to the other end segment L_2. Only when $\boldsymbol{e} \parallel \boldsymbol{e}_{L1} \parallel \boldsymbol{e}_{L2}$, there can be a curved surface without planar parts, which connects the end sections L_1, L_2. For this reason, the cylindrical bends are called from here on, *parallel bends*.

When the generator line is not parallel to the end segment ($\boldsymbol{e}_{L1} \nparallel \boldsymbol{e}$), flat surface of the elemental plate has to be extended. Therefore the elemental plate needs to be extended at least so much that there is a section parallel to the vector $\boldsymbol{e}$, or the bending axis of the parallel bend. This additional flat area added to the elemental plates to align generator lines for fitting is called *plate extension* in the context of this thesis. The problem of

finding the *minimum plate extension* is discussed later in this thesis. Then flatness is no longer required form the bend surface. It is assumed that with cylindrical bends that the bend axis and end segments are parallel.

Because of the parallelism, problem of fitting the ends of cylindrical bends can be reduced to a two-dimensional problem. There are many different ways of choosing $\boldsymbol{c}$ that will result in the same surface, but the curve with the shortest perimeter is at the surface's intersection with the plane around $\boldsymbol{e}$.

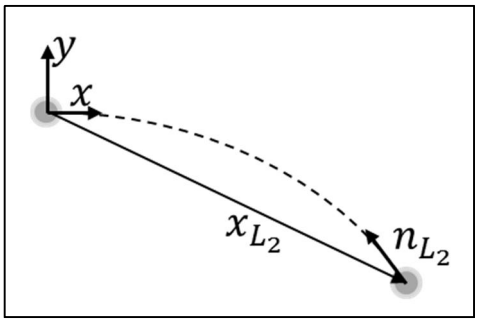


Figure 12. The two-dimensional problem of fitting a parallel bend.

The coordinate system for examining the two-dimensional problem is selected so, that the origin of the two dimensional coordinate system lies at $\boldsymbol{x}_{L1}$, and the x-direction is defined by the normal $\boldsymbol{n}_{L_1}$ of the same end segment. The y-direction is defined by the cross product $\boldsymbol{e} \times \boldsymbol{n}_{L_1}$. The z-direction is not needed in the two dimensional case, but it is defined by the vector $\boldsymbol{e}$. The coordinate transformation matrix is

$$A = \begin{bmatrix} (\hat{n}_{L_1})^T \\ (\hat{e} \times \hat{n}_{L_1})^T \\ (\hat{e})^T \end{bmatrix}^{-1}. \tag{21}$$

The two-dimensional fitting problem can then be formulated as follows:

Find such a curve $\boldsymbol{\gamma}(u), \gamma \in \mathbb{R}^2$ that fulfills the following conditions:

1. $\exists u_0, u_1 \in \mathbb{R}$, so that $\boldsymbol{\gamma}(u_0) = 0, \boldsymbol{\gamma}(u_1) = \boldsymbol{x}_{L_2}$
2. $\boldsymbol{\gamma}(u_0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$
3. $\boldsymbol{\gamma}(u_1) = \boldsymbol{n}_{L_2}$.

Several different curve parameterizations can fulfill these conditions. A circular shape is often studied in literature (Yeh et al., 1995; Wang, 1997; Eigensatz et al., 2010). However, especially with free-formed surfaces, also spline-shaped curves are often used (Zhang et al., 2007; Cai et al., 2012; Pottmann et al., 2008). When free-formed (double curved, non-developable) surfaces are approximated by semi-discrete surfaces consisting of strips of developable surfaces, the spline shape is used (Eigensatz et al., 2010). When using more advanced metal bending techniques for thin metal plates and the metal is bent in two dimensions to get a double curved surface, splines are used due to their ability to interpolate well complex geometries (Zhang et al., 2007; Cai et al., 2012).

Let us consider a circle to be fit. It is generally well known that two points and a tangent vector at the other point define a circle. In this case, there are two points and tangent vectors for both points given. Thus adding a circular bend requires a special condition, that the second tangent vector indeed is the tangent vector of the circle defined by the two points and one tangent vector. However, in the case of our problem, it is possible that the plate is extended. Therefore we can "move" the other point in the positive direction of its normal.

The condition stating that a circle can be fitted between two points can be rephrased in the form: The distance of the intersection point of the tangent lines of the end points from both of the end points, need to be the same. Figure 13 Illustrates this.

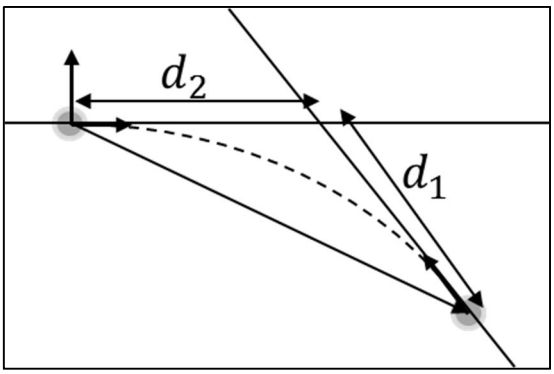


Figure 13. A circle can be fitted if d_1 and d_2 are the same length.

The approach of adding material to the flat plate works, if both d_1 and d_2 are positive. The approach works also if they are both negative, but the length of the arc could be really

long, and it could be questioned whether that is really what the user wants to do, or whether a circular bend would be the best parameterization in that case. However, if the aim is to create, for example, a pipe or a half-pipe using a long circular parallel bend, the case of both negative distances would be justified. If the distances d_1 and d_2 have different signs, a circle cannot be fit.

4.1.2 MINIMUM PLATE EXTENSION WITH A CIRCULAR PARALLEL BEND

Whenever the edges to be fit are not parallel, they can be made parallel by extending either plate (or both of them). The problem is therefore to find the "best" possible edges of both of the planes. What is considered "good" has no unambiguous answer. Possible choices would be to either maximize the distance between the new parallel edges or to minimize the amount of added material. The latter is selected, because it is a more robust definition, and in the case of parallel bends and the edges facing each other's normals, they are the same.

If there are parallel lines on two non-parallel planes, the lines have the same direction. Thus if we select any point of the intersection line of the two planes as the starting point of the parallel line, the line defines the intersection line. Therefore the parallel lines need to be also parallel to the intersection line of the two planes. The direction vector of the intersection line is denoted by $\boldsymbol{d}$.

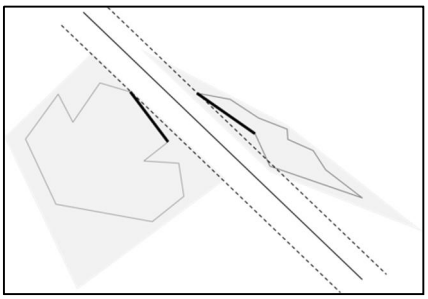


Figure 14. Minimum plate extension

The line defining the border of the minimum extension goes through the other of the end points of either of the edges – the "outer" point, as can be seen in Figure 14. Note that the "outer" edge cannot be simply defined as the edge closer to the intersection line, because in the case of a general parallel bend there is nothing that prevents the intersection line being on the other side of the edge. Therefore, if the selected edges are such that the elemental plate border segment normal vectors are pointing away from the intersection

line of the planes, the "outer" direction is away from the intersection line. The minimum plate extension borders are therefore lines that are an equal distance from the intersection line as the "outermost" point. Distance vector of a point ($\boldsymbol{y}$) from a line ($\boldsymbol{x}(t) = \boldsymbol{x}_0 + t\boldsymbol{d}$) can be computed as follows:

$$\text{dist}(\boldsymbol{x}(t), \boldsymbol{y}) = (\boldsymbol{x}_0 - \boldsymbol{y}) - \big((\boldsymbol{x}_0 - \boldsymbol{y}) \cdot \boldsymbol{d}\big)\boldsymbol{d}. \tag{22}$$

4.1.3 General developable surfaces

Let us consider the situation where two edges are connected using a developable surface. Because it is assumed that the bend is a developable surface, it can be parameterized as a ruled surface.

$$\boldsymbol{x}(u, v) = \boldsymbol{c}(u) + v\boldsymbol{e}(u). \tag{23}$$

Let the end section L_i be parameterized.

$$\boldsymbol{x}_{Li}(t_1) = \boldsymbol{x}_{0,Li} + t_1\boldsymbol{e}_{Li}, \tag{24}$$

where $i \in \{1,2\}$ depending on which end it is. The normal vectors of the elemental plate edge should lie on the tangent plane of the ruled surface, i.e. the normal vector of an elemental plate edge should be perpendicular to the normal vector of the surface at any point that belongs to the edge

$$\big(\boldsymbol{c}'(u) \times \boldsymbol{e}(u) + v\boldsymbol{e}'(u) \times \boldsymbol{e}(u)\big) \cdot \boldsymbol{n}_{L_i} = 0 \tag{25}$$

$$\big(e'(u) \times \boldsymbol{e}(u)\big) \cdot \boldsymbol{n}_{L_i} = 0. \tag{26}$$

This means that if a point belongs to L_i, there are u and v that fulfill the conditions above. Especially (26) means that for such a point, e(u), $\dot{e}(u)$ and $\boldsymbol{n}_{L_i}$ are on the same plane. Therefore $\boldsymbol{e}$ is on the plane defined by the line section direction $\boldsymbol{e}_{Li}$ and $\boldsymbol{n}_{L_i}$ for all points that belong to L_i.

The same that applies to a parallel bend; applies it also to any developable surfaces – the fitted surface around the end segment is locally a plane or $\boldsymbol{e} \parallel \boldsymbol{e}_{Li}$. Therefore, if the

amount of plate extension is to be minimized, the surface has to be selected so that $\boldsymbol{e} \parallel \boldsymbol{e}_{Li}$ at the end section.

4.1.4 CONICAL BENDS

A conical surface is a developable surface that has a singularity point at x_s. The ruled surface parameterization as in (6) is used. The generator lines all go through the one singularity point so the function $\boldsymbol{e}(u)$ can be defined as the function pointing towards the singularity point

$$\boldsymbol{e}(u) = x_s - \boldsymbol{c}(u). \tag{27}$$

The conical surface can then be written as follows

$$\boldsymbol{x}(u, v) = v\boldsymbol{x}_s + (1 - v)\,\boldsymbol{c}(u). \tag{28}$$

When fitting a conical bend, the location of the singularity point is a variable whose value is to be calculated. The result for general developable surfaces that states that the generator lines have to be parallel to fitted edges, is also required for a conical bend

$$\boldsymbol{x}_s - \boldsymbol{c}(u_i) = \alpha_i \boldsymbol{e}_{Li}, i \in \{1,2\}, \tag{29}$$

where α_i is a constant. Solving x_s and applying for both $i = 1$ and $i = 2$, we get

$$\alpha_1 \boldsymbol{e}_{L1} - \alpha_2 \boldsymbol{e}_{L2} = \boldsymbol{c}(u_2) - \boldsymbol{c}(u_1). \tag{30}$$

As we are now examining the case $\boldsymbol{e} \parallel \boldsymbol{e}_{Li}$, the directrix curve is on the end segment when $u = u_i$. The directrix curve can be selected so, that it will cross the generator line at any given point, for instance the start point of the section L_i. Therefore (29) and (30) imply that a conical surface can be fitted between two sections, if there is a plane on which the both sections and the singularity point reside.

4.1.5 MINIMUM PLATE EXTENSION FOR A CONICAL BEND

The singularity point $\boldsymbol{x}_s$ has to be located on the plane of the polygon, because the generator lines of the surface at the end segments are also on the plane. Thus, the singularity point has to be on the intersection line of the planes

$$\boldsymbol{x}_s = \boldsymbol{x}_{int} + \beta \boldsymbol{d}_{int}. \tag{31}$$

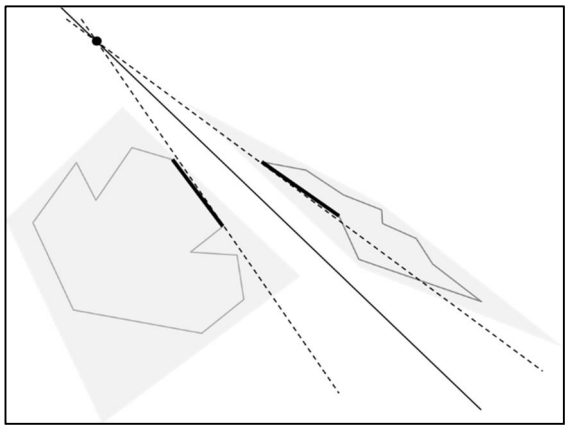


Figure 15. Finding the minimum plate extension for a conical bend

What to minimize is a lot less obvious in the case of a conical bend compared to a parallel bend, where there was an unambiguous relation between the area and the one-dimensional distance from the intersection line. In order to minimize the actual area of the plate extension, one would have to know the boundaries of the extension. For simplicity, however, the area of the triangle restricted by the end section and an equal length line section along the corresponding generator line is used. The area can be calculated using the magnitude of the cross product

$$A_i = \|\hat{\boldsymbol{e}}_{Li} \times \hat{\boldsymbol{e}}(u_i)\, \|\boldsymbol{e}_{Li}^2. \tag{32}$$

By applying (27) and (31), the area can be expressed as follows

$$A_i = \left\| \hat{\boldsymbol{e}}_{Li} \times \frac{\boldsymbol{x}_{int} + \beta \boldsymbol{d}_{int} - \boldsymbol{c}(u_i)}{|\boldsymbol{x}_{int} + \beta \boldsymbol{d}_{int} - \boldsymbol{c}(u_i)|} \right\| \; \boldsymbol{e}_{Li}^2. \tag{33}$$

In this minimization problem the only variable is β. The minimization problem can be expressed as

$$\min_{\beta} \sum_{i=1}^{2} \left(\left\| \hat{\boldsymbol{e}}_{Li} \times \frac{\boldsymbol{x}_{int} + \beta \boldsymbol{d}_{int} - \boldsymbol{c}(u_i)}{|\boldsymbol{x}_{int} + \beta \boldsymbol{d}_{int} - \boldsymbol{c}(u_i)|} \right\| \boldsymbol{e}_{L1}^2 \right). \tag{34}$$

This minimizes the area for a general cone – of whatever base. Also, because a curved section between two elemental plates is being fitted, normal vectors of the cone surface should equal to the elemental plate normal at the edge of the plate extension.

Let us now assume that the conical curved section follows a right circular cone shape. Because of the condition requiring that the surface normals be parallel to the plane normal at the edge of the plate extension, fitting a right circular cone is simplified quite a bit compared to the general case. It is already known, that the cone apex lies on the intersection line of the planes. Also it is known that the base circle tangent is on the plane on both sides. This is illustrated in Figure 16.

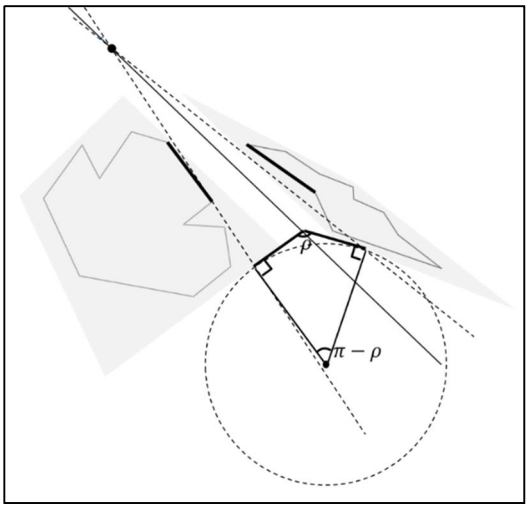


Figure 16. Fitting a circle based conical bend

Because of this, only the apex location on the intersection line and the slope of the cone define unambiguously a right conical bend surface. When minimizing the plate extension area, the expression for plate extension area given in (32) can no longer be used, because the plate extension is not a triangle on both sides anymore. Therefore, to calculate the plate extension area, we assume that the lateral edge of the plate extension is perpendicular to the intersection line. By converting the polygon points to an orthonormal coordinate system whose x-axis is parallel to the intersection line, the plate extension area is easy to calculate. This is illustrated in Figure 19. The problem of minimizing the plate extension of a circle based conical bend between two edges can therefore be expressed as follows

$$\begin{aligned} &\min_{x_s,\beta} A_1 + A_2 \\ &s.t. y_k^i - \beta\left(x_{k_i}^i - x_s\right) \geq 0, \forall i \in \{1,2\}, \end{aligned} \tag{35}$$

where k_i is the selected index of the edge on side i, and the areas on each side are

$$\begin{aligned} A_1 &= \left(x_{k_1+1}^1 - x_{k_1}^1\right)\left(½\left(y_{k_1+1}^1 + y_{k_1}^1\right) - ½\beta\left(x_{k_1+1}^1 + x_{k_1}^1 - 2x_s\right)\right) \\ A_2 &= \left(x_{k_2+1}^2 - x_{k_2}^2\right)\left(½\left(y_{k_2+1}^2 + y_{k_2}^2\right) - ½\beta\left(x_{k_2+1}^2 + x_{k_2}^2 - 2x_s\right)\right). \end{aligned} \tag{36}$$

The decision variables are the location of the apex x_s and the line slope β. This is a non-linear optimization problem, because the decision variables are multiplied. However, the decision variable x_s is only present in terms where it is multiplied by the variable β. When chosen the variables β and $x_\beta = \beta\, x_s$ as decision variables, the optimization problem can be treated as a linear optimizaton problem that can be solved fast and reliably. Because the term $\left(y_{k_1+1}^1 + y_{k_1}^1\right)\left(x_{k_1+1}^1 - x_{k_1}^1\right)$ is a constant, it can be removed from the objective function. The linear optimization formulation of the plate extension minimizing problem is

$$\begin{aligned} &\min_{x_\beta,\beta} \frac{1}{2}\left(\Delta x_{k_1}\left(x_{k_1+1}^1 + x_{k_1}^1\right)\beta - 2\Delta x_{k_1} x_\beta + \Delta x_{k_2}\left(x_{k_2+1}^2 + x_{k_2}^2\right)\beta\right. \\ &\qquad\qquad \left. - 2\Delta x_{k_2} x_\beta\right) \\ &s.t. y_k - \beta x_{k_i}^i + x_\beta \geq 0, \forall i \in \{1,2\}, \end{aligned} \tag{37}$$

where $\Delta x_{k_i} = \left(x_{k_i}^i - x_{k_i+1}^i\right)$ is a constant. This can either be solved analytically or by using linear programming.

4.1.6 MANY-TO-MANY EDGES BEND FITTING PROBLEM

Often the user might want to connect several edges from both sides as is illustrated in Figure 17. More than one consecutive edges on one plane are to be connected to another array of consecutive edges on another plane using a bend.

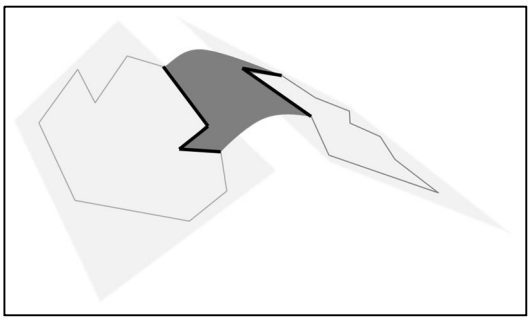


Figure 17. Many-to-many bend fitting

The plane to be fit is parameterized as in (23), the end sections are parameterized as in (24), but $1 < i < j + k \mid j, k \in \mathbb{Z}^+$ because there can be any number of edges. It is assumed that the first j edges are the edges of one side and the next k edges are the edges on the other side. Equation (26) is also valid for this case.

Because the normal vectors $\boldsymbol{n}_{L_i}$ on one side are all on the same plane as the corresponding polygon, $\boldsymbol{e}$ has to be on that same plane whenever u and v are such that $\boldsymbol{x}(u, v) \in L_i$, where i is any index of an edge on one side. Because a ruled surface consists of generator lines, there has to be a set of lines on the surface that encloses all of the points of the edges on either side.

Therefore the many-to-many edges bend fitting problem is also reduced to the problem of finding the appropriate plate extension and connecting two edges with a developable surface. For a circular parallel bend finding the minimum plate extension is easy. The problem setting is analogous to the case of connecting two single edges. Minimum plate extension edge is found by finding the polygon point with a minimum distance from the intersection line of the planes, and setting the connecting edge at that distance on both sides.

Many-to-many bend fitting problem is more complicated for non-parallel bends, because the bend boundaries affects the area of plate extension. To simplify the problem, the area of the plate extension is bounded by a line normal to the intersection line of the planes as was done in Section 4.1.5. This is illustrated in Figure 18.

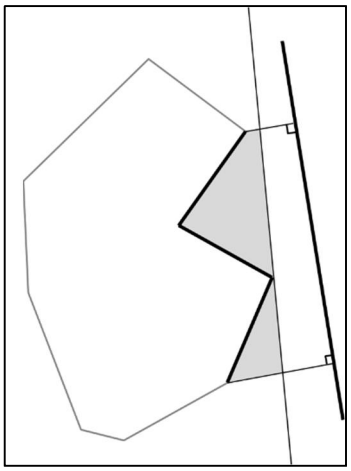


Figure 18. Minimum plate extension of a conical bend

As in Section 4.1.4, the boundary line between the plate extension and the curved section is a line that goes through the apex of the cone, which lies on the intersection line of the planes. If the apex is fixed, the boundary line goes through (at least) one of the corner points of the selected edges on one of the sides. The edge through which it goes, is the one that minimizes the plate extension area. Selecting any other point would obviously cause the elemental plate to be cut, which is not permitted in this thesis. The minimum plate extension for the other side of the bend is found by defining the boundary line so, that it also goes though the apex and has the same slope. Then by varying the location of the apex, the overall minimum plate extension is found. A similar approach as in Section 4.1.5 is used for solving the minimum plate extension.

4.1.7 SOLVING MANY-TO-MANY FITTING PROBLEM USING LINEAR PROGRAMMING

The points on both sides are converted into a local two-dimensional orthonormal coordinate system whose x-axis is parallel to the intersection line. Let the x-coordinate of the k_ith corner point of the polygon on side i be $x_{k_i}^i$. Let the corresponding y-coordinate be $y_{k_i}^i$. The coordinate systems on both sides are selected so, that their origin is located at the same point in the 3-dimensional space. The total plate extension area is minimized, requiring that the y-value y_k^i for each point never exceeds $\beta(x_k^i - x_s)$.

$$\begin{aligned} &\min_{x_s,\beta} A_1 + A_2 \\ &s.t. y_k - \beta\left(x_k^i - x_s\right) \geq 0, \forall i \in \{1,2\}, k \in \mathbb{N}, k \leq n_i, \end{aligned} \tag{38}$$

where

$$A_1 = \sum_{k=1}^{n_1} (x_{k+1}^1 - x_k^1)\left(½(y_{k+1}^1 + y_k^1) - ½\beta(x_{k+1}^1 + x_k^1 - 2x_s)\right)$$
$$A_2 = \sum_{k=1}^{n_2} (x_{k+1}^2 - x_k^2)\left(½(y_{k+1}^2 + y_k^2) - ½\beta(x_{k+1}^2 + x_k^2 - 2x_s)\right). \quad (39)$$

There are several algorithms for solving linear optimization problems. One of the most used algorithms is the Simplex algorithm. Computational efficiency of the simplex algorithm, depends on the computational efficiency of an iteration and the number of iterations. One iteration can be solved in $O(m\,n)$ time. On average it takes around $O(m)$ iterations to calculate a linear programming problem. In the worst case, it requires $O(2^n)$ iterations. Here m stands for the number of conditions and n for the number of variables. (Bertsimas & Tsitsiklis, 1997)

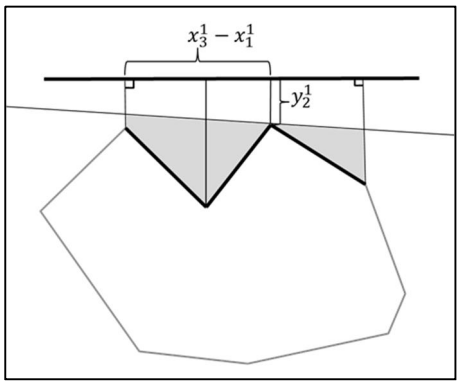


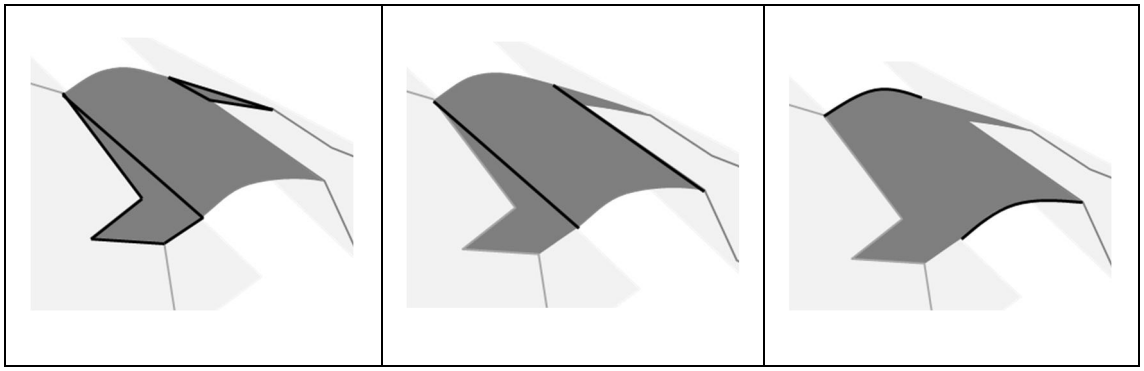
Figure 19. Calculating the area of the minimum plate extension in the many-to-many conical bend fitting problem

The above formulation has a relatively small number of both constraints and variables. Therefore, it is justified to use linear programming to solve the optimization problem. The optimization problem can be formulated to be a linear optimization problem in the same way as in Section 4.1.5.

This formulation only works if all the edges are facing the intersection line. If the chosen edges are facing away from the intersection line, a similar optimization problem cab solved for edges facing away from the intersection line.

4.2 BEND BOUNDARIES

The curvature properties of a curved section and curved section geometric properties have been discussed in Section 2.4 without defining the boundaries of the bend. The boundaries of a bend can be roughly divided into two main categories, boundaries of the plate extension and the boundaries of the curved section. The boundaries of the curved section can further be divided into two, the boundary along the end segments of the curved section (and the generator lines of the surface), and the boundaries on the sides of the bend. The different boundary types are illustrated in Figure 20.



(a) (b) (c)

Figure 20. The different boundary types of a bend

The boundaries along the end sections as illustrated in Figure 20.b, have already been discussed and the location of the line which it follows is found out as a result of the minimum extended plate -problem. Its length and end points have not yet been discussed. Let us call this boundary the *discontinuity boundary* due to the fact that at this boundary, the curved section is discontinued and a plane section of the bend begins.

Boundaries of the plate extension as illustrated in Figure 20.a include the same boundary found in the minimum extended plate -problem. Because the extended plate is a possibly concave polygon on a plane, the ordered set of all polygon points have to be known. If either of the end points of the end section do not lie on the polygon points of the corresponding polygon, at least one boundary edge is needed to connect the original polygon edges to the discontinuity boundary.

The curved section is bounded by the discontinuity boundary and the lateral boundaries as illustrated in Figure 20.c. A curved section is fully defined by its boundaries and the parameters of the surface. The lateral boundaries can be of any shape, and it will not affect the developability of the surface, but it may affect the global geometric unfoldability of

the surface. For example, if the lateral edge in Figure 20.c was shifted towards the viewer, it could be on the plate on the right side, thus causing the whole bend not to be unfoldable. This is illustrated in Figure 21.

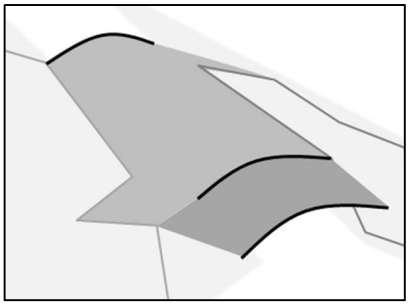


Figure 21. A legal unfoldable bend (lighter) and an illegal bend (darker) that makes the object not unfoldable.

4.3 LATERAL BOUNDARIES

To prevent the situation in Figure 21, and to make sure that the entire object stays globally geometrically unfoldable, the lateral boundaries have to be selected in such a way, that no inner points of the polygon belong to the curved section. However, that is not a sufficient condition for unfoldability, but is also needed. To make sure that such points do not exist, one can find all the points where the line following the end section intersects with the polygon. The sections of the line for which there are inner polygon points, are illegal sections. All other sections are legal. A bend end section will have to be fully on a legal section. An algorithm for finding the overlapping sections between a line and polygon is presented in Appendix 1.

It is assumed now that two polygonal plates are fitted together with a bend, and a set of consecutive edges is selected on both sides. One could also assume that the user is not willing to extend the curved section beyond the selected elemental plate border segments, thereby defining the span of the curved section through selecting edges. Thus, if an extreme polygon end points on one side lies on the discontinuity boundary, that point can be considered a limit for the lateral boundary. If there is a plate extension present, it is assumed that the lateral boundary of the curved section is in line with the plate extension's added edge. Therefore, if the corner points of the plate extension on the discontinuity boundary are known, then the lateral boundaries will intersect those points.

There are several desirable properties the geometry of the lateral boundary (possibly together with the extended section) could fulfill. The geometric properties of the curve bounding the curved section on the lateral side could be selected in such a way that they are (a) somehow optimal, (b) easy to understand or model, or (c) provide the user with a powerful tool to model curved geometries.

One natural restriction for the shape of the lateral boundary is that there should be a unique value for the parameter v in (6) for each point on the boundary, so that the lateral boundary could be simply be defined by varying the lengths of the generator lines. By assuming this, one can use the following approach to study the shape of the boundary both when unfolded and when not.

4.4 UNFOLDING A DEVELOPABLE SURFACE

Unfolding a developable surface is essentially about finding a 1-to-1 bijective relationship between the surface and a planar coordinate system. It is also required that distances are preserved. Therefore in order to unfold a surface, one has to find an isometric mapping from the surface to a plane.

The geodesics of a developable surfaces are straight line segments after unfolding (Bo & Wang, 2007). The generator lines are therefore also geodesics. One can imagine a coordinate system at a given point (the origin) on the unfolded surface. The x-coordinate would be along a geodesic (other than a generator line). The angle between the geodesic and a generator line going through the origin can be calculated. Based on this information the location of the extreme point of the surface on that generator line can be calculated also in the two dimensional coordinate system. By going from the geodesic along the generator line to the edge always knowing that the geodesic lies on the x-axis, one can derive the coordinate values of the extreme points of the bent section. Based on this idea Clements & Leon (1987) suggest a method for calculating the unfolded geometry:

A curve $\boldsymbol{g}(s)$ on a developable surface $\boldsymbol{S}$ is a geodesic, if the plane of curvature of $\boldsymbol{g}(s)$ is perpendicular to the tangent plane of $\boldsymbol{S}$ on all points in $\boldsymbol{g}$. This requirement can be expressed in the following form

$$\boldsymbol{g}'' \cdot (\boldsymbol{g}' \times \boldsymbol{n}) = 0. \tag{40}$$

Also there is a function $v^*(u)\colon \mathbb{R} \to \mathbb{R}$, and a geodesic on a developable surface, but not on its generator line, that can be formulated as follows based on the ruled surface parameterization in (6)

$$\boldsymbol{g}(u) = \boldsymbol{c}(u) + v^*(u)\boldsymbol{e}(u). \tag{41}$$

By calculating $\boldsymbol{g}''$ and $\boldsymbol{g}'$ based on (41) and applying the resulting equations to (40), a second order non-linear differential equation is revealed. The derivatives of g are the following

$$\begin{aligned} \boldsymbol{g}'(u) &= \boldsymbol{c}'(u) + v^*(u)\boldsymbol{e}'(u) + v^{*\prime}(u)\boldsymbol{e}(u) \\ \boldsymbol{g}'' &= \boldsymbol{c}'' + v^*\boldsymbol{e}'' + 2v^{*\prime}\boldsymbol{e}' + v^{*\prime\prime}\boldsymbol{e}. \end{aligned} \tag{42}$$

The differential equation can further be formed into a system of two first order differential equations. The differential equation system is then

$$\begin{cases} x_1' = x_2 \\ x_2' = -\dfrac{\boldsymbol{c}'' \cdot (\boldsymbol{g}' \times \boldsymbol{n})}{\boldsymbol{e} \cdot (\boldsymbol{g}' \times \boldsymbol{n})} - \dfrac{\boldsymbol{e}'' \cdot (\boldsymbol{g}' \times \boldsymbol{n})}{\boldsymbol{e} \cdot (\boldsymbol{g}' \times \boldsymbol{n})}\, x_1 - 2\dfrac{\boldsymbol{e}' \cdot (\boldsymbol{g}' \times \boldsymbol{n})}{\boldsymbol{e} \cdot (\boldsymbol{g}' \times \boldsymbol{n})} x_2 \end{cases}, \tag{43}$$

where $x_1 = v^*(u)$. The variables $\boldsymbol{c}$, $\boldsymbol{e}$, $\boldsymbol{g}$ and $\boldsymbol{n}$ are all functions of the variable u, but a simplified notation is now used. Clements & Leon (1987) parameterized the ruled surface in such a way that $0 \leq v^* \leq 1$, so that the boundaries of the developable surface can be found by setting v^* to zero or one. After all, when unfolding a developable surface one is most interested in the shape of the lateral boundaries of the curved section.

4.5 SELECTING THE LATERAL BOUNDARY

The user may want to create such a bend that the unfolded plate is the form of a polygon. Fabrication of a polygonal steel plate is considerably more straightforward than producing a steel plate with round edges. Therefore a situation is considered where the bent section lateral boundary is a straight line in the unfolded state. The cases of a circular parallel bend and a right circle-based conical bend are considered. Using the approach presented in 4.4, the shape of the boundary in three-dimensional space is derived.

4.5.1 PARALLEL BEND STRAIGHT LINE LATERAL BOUNDARY

The parallel bend is parameterized into a form where the independent variable of the generator line is assumed to vary between zero and one, thus specifying the shape of the lateral boundary. A parallel bend for surface unfolding is formulated as

$$\boldsymbol{x}(u,v) = \boldsymbol{c}(u) + (\omega(u) + v\lambda(u))\boldsymbol{e_0}\,, \tag{44}$$

where $0 \leq v \leq 1, \lambda(u), \omega(u) \in \mathbb{R}$. The function $\omega(u)$ determines the location of one lateral boundary expressed as the distance from the directrix curve. The sum of the functions $\omega(u)$ and $\lambda(u)$ define the location of the other lateral boundary. It is also assumed that $\boldsymbol{e}_0$ is normalized. Because the generator line directions stay the same for a parallel bend, the normal of the surface can be simplified to the following form

$$\boldsymbol{n}(u) = \boldsymbol{c}'(u) \times \boldsymbol{e_0}. \tag{45}$$

From Equations (45) and (40) it can be concluded, that if c is a circle, it is in fact a geodesic of the surface because $\boldsymbol{c}'' \times (\boldsymbol{c}' \times \boldsymbol{e}) = \boldsymbol{c}'(\boldsymbol{c}'' \cdot \boldsymbol{e}) - \boldsymbol{e}(\boldsymbol{c}'' \cdot \boldsymbol{c}') = 0\,\boldsymbol{c}' - \,0\,\boldsymbol{e} = 0$. Therefore, the problem is then trivial and $\boldsymbol{g}(u) = \boldsymbol{c}(u)$. Any parameterization for a circle can then be used for the geodesic. The x-coordinate value of the lateral boundary of a circular bend is then calculated simply by integrating the directric curve, which is now also the geodesic

$$x_{2D}(u) = \int_0^u \|c'(t)\| dt. \tag{46}$$

If the circle $\boldsymbol{c}(u)$ is parameterized as $\boldsymbol{c}(u) = \boldsymbol{\alpha} + R(\cos(u)\,\boldsymbol{r}_0 + \sin(u)\,\boldsymbol{e} \times \boldsymbol{r}_0)$, where r_0 is the normalized radius vector of the circle at the beginning of the arc length. Therefore, because u is representing the angle of the circle section (in radians), the length can simply be calculated as

$$x_{2D}(u) = Ru. \tag{47}$$

Because $e_0 \perp c'(u)$, one can directly derive from Equation (44) an expression for the y-coordinate on the unfolded plane

$$y_{2D}(u) = \omega(u) + v\lambda(u). \tag{48}$$

At the first lateral boundary the v-coordinate is zero and the u-coordinate can be calculated using (47) and (48)

$$\begin{cases} v = 0 \\ u = \frac{x_{2D}}{R}. \end{cases} \tag{49}$$

If the first lateral boundary of the unbent section were a straight line other than the obvious y_{2D} equals to a constant, x_{2D} would be a linear function of y_{2D}

$$u = \frac{\alpha_0 y_{2D} + \beta_0}{R}. \tag{50}$$

Based on this one can derive a formula for $\omega(u)$

$$\omega(u) = y_{2D} = \frac{Ru - \beta_0}{\alpha_0}. \tag{51}$$

Similarly for the lateral edge on the other side ($v = 1$), the relationship between the x, y –coordinates on the unfolded plane and the u, v –coordinates of the surface in space can be used to calculate

$$\lambda(u) = \frac{Ru - \beta_1}{\alpha_1} - \frac{Ru - \beta_0}{\alpha_0}. \tag{52}$$

Thus the surface of a circular parallel bend that has a linear edge in the unfolded state, can be parameterized as

$$\boldsymbol{x}(u, v) = \alpha + R\cos(u)\, r_0 + R\sin(u)\, e \times r_0 + ((1 - v)\frac{Ru - \beta}{\alpha_0} + v(\frac{Ru - \beta_1}{\alpha_1}))\boldsymbol{e_0}. \tag{53}$$

Note that this result is only valid for a circular parallel bend. The problem is not complicated for other parallel bends either, if the generator lines are normal to the directrix curve, because the directrix curve is then a geodesic.

4.5.2 STRAIGHT LINE LATERAL BOUNDARY ON CONICAL, NON-PARALLEL BENDS

Unfolding a conical shape is considerably more difficult than unfolding a cylindrical shape, because the geodesics will not follow the directrix curve, and therefore the geodesic has to be found, and only then derive the location of a point on the unfolded plane by utilizing the geodesic. Let a circle based cone surface be expressed using cylindrical coordinates

$$\boldsymbol{s}(\theta, R) = \boldsymbol{s}_0 + R(\boldsymbol{r}_0 \cos\theta + \boldsymbol{r}_0 \times \boldsymbol{h}_0 \sin\theta) + mR\boldsymbol{h}_0 \,, \tag{54}$$

where m is the steepness of the cone (radius/height), $\boldsymbol{r}_0$ is a unit vector on the cone base, $\boldsymbol{h}_0$ is the direction vector of the cone axis, s_0 is the location of the apex of the cone, h is the height of the cone. The variables are the radius of the base circle R, and the azimuth angle θ. The lateral boundaries can simply be expressed as boundary values for R as a function of θ

$$R_{min}(\theta) \leq R \leq R_{max}(\theta). \tag{55}$$

Instead of trying to solve the differential equation system in (45), which would form a relatively complicated problem that would be difficult to solve analytically, the geodesic of a right circular cone is found using calculus of variations. It is proven in Appendix 2, that geodesic of a right circular cone needs to satisfy the condition

$$R\cos\left(\frac{\theta + C_1}{\sqrt{1+m^2}}\right) = C_2. \tag{56}$$

Because this result only applies to a right circular cone, it is assumed from here on, that conical bends consist of curved sections following the surface of a right circular cone. The geodesic between two points on a right circular cone can be found by solving the constants C_1 and C_2. Because the discontinuity boundaries of a conical bend are both the same angle θ distance from the intersection line as shown in Figure 16, the vector $\boldsymbol{r}_0$ is chosen so that it points towards the intersection line. Let $(-\theta_0, R_1)$ and (θ_0, R_2) be the coordinates of the points between which the geodesic is found. Based on Appendix 2, C_1 and C_2 are

$$C_1 = \sqrt{1+m^2}\tan^{-1}\left(\frac{R_2 - R_1}{R_2 + R_1}\cot\frac{\theta_0}{\sqrt{1+m^2}}\right)$$
$$C_2 = R_1\cos\left(\frac{-\theta_0 + C_1}{\sqrt{1+m^2}}\right). \quad (57)$$

The aim here is to parameterize the geodesic curve using one variable. Variable R can then be expressed as a function of the variable θ and the geodesic can be parameterized using (56) as a curve of a parameter θ so that

$$\boldsymbol{s_{gd}}(\theta) = \boldsymbol{s}_0 + R(\theta)(\boldsymbol{r}_0\cos\theta + \boldsymbol{r}_0 \times \boldsymbol{h}_0\sin\theta) + aR(\theta)\boldsymbol{h}_0, \quad (58)$$

where the function $R(\theta)$ is derived from (56)

$$R(\theta) = \frac{C_2}{\cos\left(\frac{\theta + C_1}{\sqrt{1+m^2}}\right)}. \quad (59)$$

In addition to the formula for the geodesic, we need to know its length from the origin $(\theta_1, R(\theta_1))$ to an arbitrary point $(\theta_L, R(\theta_L))$

$$L(\theta_L) = \int_{\theta_0}^{\theta_L}\|d\boldsymbol{s}_{gd}\| = \int_{\theta_1}^{\theta_L}\sqrt{(1+m^2)\left(\frac{dR}{d\theta}\right)^2 + R^2}. \quad (60)$$

Because the curve is presented as a function of θ, the integration variable has to be changed from R to θ, which gives

$$\frac{dR}{d\theta} = \frac{C_2\tan\left(\frac{C_1+\theta}{\sqrt{1+m^2}}\right)}{\cos\left(\frac{C_1+\theta}{\sqrt{1+m^2}}\right)\sqrt{1+m^2}}. \quad (61)$$

Using (59) – (61) to solve the integral given the following expression for the curve length of the geodesic of a right circular cone, we get

$$L(\theta_L) = C_2\sqrt{1+m^2}\left(\tan\left(\frac{C_1+\theta_L}{\sqrt{1+m^2}}\right) - \tan\left(\frac{C_1-\theta_0}{\sqrt{1+m^2}}\right)\right). \quad (62)$$

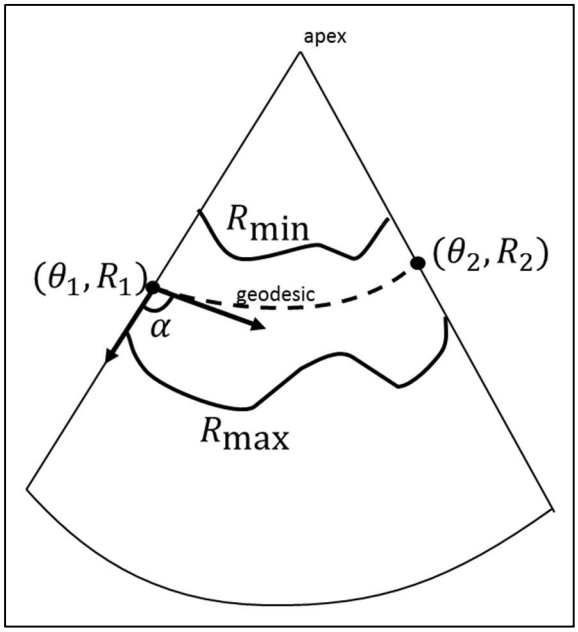


Figure 22. The lateral boundaries and the geodesic in unfolding a right circular cone.

Now the position of the reference point is known. In (58) the generator lines for any point can be found by simply fixing the value of θ. Thus the generator line for the reference point can be found by fixing θ to θ_L. But before the location of a point on the lateral boundary can be calculated, one has to know the distance from the reference point to the edge along the generator line and the angle between the generator line and the geodesic at the reference point. A formula for the cosine and sine of the angle are derived in Appendix 3 and they are the following

$$\cos\big(\alpha(\theta_L)\big) = \frac{R'(\theta_L)}{\left(R'^2(\theta_L) + \frac{R^2(\theta_L)}{(m^2+1)}\right)^{\frac{1}{2}}} \tag{63}$$

$$\sin\big(\alpha(\theta_L)\big) = \frac{R(\theta_L)}{\big((m^2+1)R'^2(\theta_L) + R^2(\theta_L)\big)^{\frac{1}{2}}}. \tag{64}$$

Equation (63) gives the cosine of the angle, but it still needs to be known whether the angle is to the “right” or to the “left” from the x-axis. The generator lines’ direction vectors are pointing away from the apex. The geodesic is a directed curve, and the

direction of the geodesic is whatever it is chosen to be, for example from $(-\theta_0, R_1)$ to (θ_0, R_2). Therefore the angle α is the angle between the direction of the geodesic and a vector pointing towards the outer lateral boundary of the curved section $R_{\max}(\theta)$. The distance from the reference point to the outer lateral (maximum) edge along the generator line is

$$\begin{aligned}\Delta_{\max}(\theta_L) &= \left\| \boldsymbol{s}_{gd}(\theta_L) - \boldsymbol{s}\big(\theta_L, R_{\max}(\theta_L)\big) \right\| \\ &= \big(R_{\max}(\theta_L) - R(\theta_L)\big)\sqrt{m^2+1}.\end{aligned} \tag{65}$$

Knowing that the reference point lies in $(L(\theta_L), 0)$ on the unfolded plane and by using Equations (63) – (65), the x- and y- coordinates of the lateral (maximum) boundary are

$$\begin{aligned}x_{\max}(\theta_L) &= L(\theta_L) + \cos\big(\alpha(\theta_L)\big)\,\Delta_{\max}(\theta_L) \\ y_{\max}(\theta_L) &= -\sin\big(\alpha(\theta_L)\big)\,\Delta_{\max}(\theta_L).\end{aligned} \tag{66}$$

Expanding these, the expression gives

$$\begin{aligned}x_{\max}(\theta_L) &= L(\theta_L) + (m^2+1)R'(\theta_L)\frac{R_{\max}(\theta_L) - R(\theta_L)}{\sqrt{(m^2+1)R'^2(\theta_L) + R^2(\theta_L)}} \\ y_{\max}(\theta_L) &= -\sqrt{m^2+1}\,R(\theta_L)\frac{R_{\max}(\theta_L) - R(\theta_L)}{\sqrt{(m^2+1)R'^2(\theta_L) + R^2(\theta_L)}}.\end{aligned} \tag{67}$$

This set of equations describes the relationship between the coordinates of the outer lateral boundary of a conical curved section in the 3-space and on the unfolded plane. As in 4.5, the shape and parameters for a conical bend are derived for such a boundary curve that is a straight line on the unfolded plane

$$y_{\max} = A + Bx_{\max}. \tag{68}$$

Deriving the value of $R_{\max}(\theta_L)$ for a straight line lateral boundary is done in Appendix 4

$$R_{\max}(\theta_L) = \frac{C_2}{\cos\left(\frac{\theta_L + C_1}{\sqrt{1+m}}\right)} - \frac{A + BL(\theta_L)}{\left(\sqrt{m^2+1}\right)\cos\left(\frac{\theta_L + C_1}{\sqrt{1+m}}\right) + \frac{(m^2+1)}{\sqrt{1+m}}\sin\left(\frac{\theta_L + C_1}{\sqrt{1+m}}\right)B}. \quad (69)$$

The value of the radius R corresponding to the minimum inner boundary of the conical bend can be derived using a similar method. The distance $\Delta_{\min}$ is calculated as

$$\begin{aligned}\Delta_{\min}(\theta_L) &= \left\| \boldsymbol{s}_{gd}(\theta_L) - \boldsymbol{s}\big(\theta_L, R_{\min}(\theta_L)\big) \right\| \\ &= \big(R(\theta_L) - R_{\min}(\theta_L)\big)\sqrt{a^2+1}.\end{aligned} \quad (70)$$

The inner lateral boundary is towards the apex, i.e. against the direction of the generator line, so the x- and y- coordinates of the minimum lateral boundary are

$$\begin{aligned}x_{\min}(\theta_L) &= L(\theta_L) - \cos\big(\alpha(\theta_L)\big)\,\Delta_{\min}(\theta_L) \\ y_{\min}(\theta_L) &= +\sin\big(\alpha(\theta_L)\big)\,\Delta_{\min}(\theta_L)\,.\end{aligned} \quad (71)$$

Because $\Delta_{\min}$ is the same as $\Delta_{\max}$ but negative, and also $x_{\min}$ and $y_{\min}$ are of the same form as $x_{\max}$ and $y_{\max}$ except negative too, they follow the same form as (67)

$$\begin{aligned}x_{\min}(\theta_L) &= L(\theta_L) + (m^2+1)R'(\theta_L)\frac{R_{\min}(\theta_L) - R(\theta_L)}{\sqrt{(m^2+1)R'^2(\theta_L) + R^2(\theta_L)}} \\ y_{\min}(\theta_L) &= -\sqrt{m^2+1}\,R(\theta_L)\frac{R_{\min}(\theta_L) - R(\theta_L)}{\sqrt{(m^2+1)R'^2(\theta_L) + R^2(\theta_L)}}\,.\end{aligned} \quad (72)$$

4.5.3 COMPLETELY STRAIGHT LINE LATERAL BOUNDARY

In this section the objective is to find out a bend boundary that is a line in the unfolded state – so that the curved section lateral boundary continues as a straight line as the plate extension boundary.

Because the border line that connects the elemental plate polygon points is a straight line in the unfolded state, it is a straight line on the plate extension and a geodesic on the

curved section. Therefore, it is enough to find points on the bent section discontinuity boundaries such that the lines from the polygon corner points to the selected points on discontinuity boundaries have the same value of derivative as the value of the derivative of the curved section at that same point. This sounds simple, but results in a complicated system of equations that cannot be solved analytically.

Another way to solve the problem is to minimize the total length of the geodesic and the line segments. The line segments length is easy to calculate using the Pythagorean Theorem, and (62) gives the length of a geodesic. To analytically minimize the total length, the partial derivatives $\frac{\partial L_{total}}{\partial R_1}$ and $\frac{\partial L_{total}}{\partial R_2}$ need to be calculated. It is possible to find a complicated analytical solution for them, but solving them for R_1 and R_2 is difficult because of the high degree of complexity.

However, there is a simpler and more intuitive way of solving the problem. Also, the approach can be used for other developable surfaces whose geodesic can be calculated. It is possible to derive equations for the discontinuity boundary on the unfolded plane. Equally, one can easily calculate the polygon end point coordinate values on the unfolded plane. Using those results, the intersection point of a discontinuity boundary and the line between the two polygon points reveal where the straight-line lateral boundary intersects the discontinuity boundary. From there one can calculate the equivalent geodesic of a cone using (58). The process of finding the corner points of the plate extension can be done in the following steps:

1. Select end points and calculate parameters for an arbitrary reference geodesic on the cone. This will be the x-axis on the unfolded plane.
2. Calculate the length L of the reference geodesic.
3. Calculate the angle between the discontinuity boundary and the reference geodesic.
4. Let us assume that the beginning of the geodesic lies in $(0, 0)$ on the unfolded plane. Therefore the end of the geodesic lies in $(L, 0)$. Knowing these points and the angle between the x-axis and the lines, the discontinuity border lines' parameters are calculated.
5. Calculate the distance of the polygon point from the reference geodesic end point, and the angle between the distance vector and the discontinuity boundary. Based

on the point and distance, calculate the location of the polygon point on the unfolded plane.

6. Calculate at which point R on the unfolded plane of the discontinuity boundary the line between the polygon points intersects the discontinuity boundary.

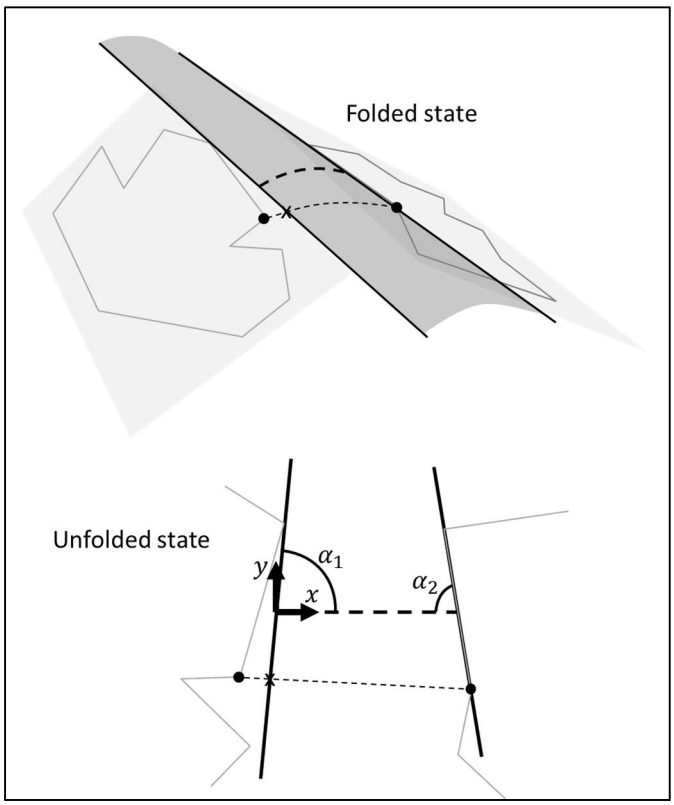


Figure 23. Finding the straight line between polygon points on different sides of the conical curved section

Step 1. Theoretically the reference geodesic could be placed anywhere except the apex of the cone. To preserve the computational precision of the calculation, it is reasonable to place the origin of the new coordinate system close to the polygon points to be connected. Therefore, let the reference geodesic end points be projections of the polygon points to the discontinuity boundaries.

The discontinuity boundary is expressed in cylindrical coordinates as

$$\begin{aligned} \boldsymbol{\delta}_1(R) &= \boldsymbol{s}_0 + R(\boldsymbol{r}_0 \cos -\theta_0 + \boldsymbol{r}_0 \times \boldsymbol{h}_0 \sin -\theta_0) + mR\boldsymbol{h}_0 \\ \boldsymbol{\delta_2}(R) &= \boldsymbol{s}_0 + R(\boldsymbol{r}_0 \cos \theta_0 + \boldsymbol{r}_0 \times \boldsymbol{h}_0 \sin \theta_0) + mR\boldsymbol{h}_0. \end{aligned} \tag{73}$$

The projection of the polygon point to the discontinuity boundary is

$$\mathrm{proj}_{\bar{\boldsymbol{\delta}}_\iota}(\bar{\boldsymbol{x}}_J) = \frac{\bar{\boldsymbol{x}}_\iota \cdot \bar{\boldsymbol{\delta}}_\iota}{\bar{\boldsymbol{\delta}}_\iota \cdot \bar{\boldsymbol{\delta}}_\iota} \bar{\boldsymbol{\delta}}_\iota, \tag{74}$$

where $\bar{\boldsymbol{x}}_J = \boldsymbol{x}_j - \boldsymbol{s}_0$, and $\bar{\boldsymbol{\delta}}_\iota = \boldsymbol{\delta}_i(1) - \boldsymbol{s}_0$, $i \in \{1,2\}$. The equivalent starting and ending points of the reference geodesic would be $s_0 + \mathrm{proj}_{\bar{\boldsymbol{\delta}}_\iota}(\bar{\boldsymbol{x}}_J)$. By first converting the polygon point to the cylindrical coordinate system of the cone, and calculating the value of the projection, we get the parameter R for the start and end points of the geodesic. The projection from (74) and the R value of the end points of the reference geodesic are therefore

$$\begin{aligned} \boldsymbol{s}_{gd,i} &= \boldsymbol{s}_0 + R_{gd,i}(\boldsymbol{r}_0 \cos\theta_i + \boldsymbol{r}_0 \times \boldsymbol{h}_0 \sin\theta_i + m\boldsymbol{h}_0) \\ R_{gd,i} &= \frac{\left(\cos\theta_i x_{\boldsymbol{r}_0,i} + \sin\theta_i x_{r_0 \times h_0,i} + m x_{h_0,i}\right)}{(1+m)}. \end{aligned} \tag{75}$$

Based on these, the parameters of the geodesic can be calculated using (57).

Step 2. The length of the geodesic can be calculated using (62).

Step 3. The cosine of the angle between the geodesic and the discontinuity boundary (as depicted as α_i in Figure 23) can be calculated using the dot product of the values of the derivatives of both curves at the intersection points. Another way to calculate the angle α_1, is to utilize the fact that the apex of the cone and the geodesic end points form a triangle whose all edge lengths are known.

Step 4. The first discontinuity boundary goes through the point $(0, 0)$ on the unfolded plane. Its slope is given by $\tan(\alpha_1) = \sqrt{1 - \cos^2(\alpha_1)} \;/\; (\cos(\alpha_1))$, because the angle α_i is positive and on top of the geodesic as described in Figure 23. The second discontinuity boundary goes through $(L, 0)$ and its slope can be calculated similarly $\tan(\alpha_2) = \frac{\sqrt{1-\cos^2(\alpha_2)}}{\cos(\alpha_2)}$.

Step 5. The distance vector is obtained by subtracting the projection vector from the polygon point. Because the geodesic start and end points are projections of the polygon points to the lines, the angle is already known. Therefore the coordinates of the point can be calculated the following way

$$
\begin{aligned}
x_1 &= -\left(\overline{\boldsymbol{x_1}} - \mathrm{proj}_{\bar{\boldsymbol{\delta}}_i}\left(\overline{\boldsymbol{x}_j}\right)\right)\cos\alpha_1 \\
y_1 &= -\left(\overline{\boldsymbol{x_1}} - \mathrm{proj}_{\bar{\boldsymbol{\delta}}_i}\left(\overline{\boldsymbol{x}_j}\right)\right)\sin\alpha_1 \\
x_2 &= L + \left(\overline{\boldsymbol{x_2}} - \mathrm{proj}_{\bar{\boldsymbol{\delta}}_i}\left(\overline{\boldsymbol{x}_j}\right)\right)\cos\alpha_1 \\
y_2 &= \left(\overline{\boldsymbol{x_2}} - \mathrm{proj}_{\bar{\boldsymbol{\delta}}_i}\left(\overline{\boldsymbol{x}_j}\right)\right)\sin\alpha_1 .
\end{aligned}
\tag{76}
$$

Step 6. On the first side, the line connecting the points should be equal to the discontinuity boundary

$$
\delta_1 \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + (1-\delta_1)\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \gamma_1 \begin{bmatrix} 1 \\ \tan(\alpha_1) \end{bmatrix}. \tag{77}
$$

By solving this for the line parameters δ_1 and γ_1, we get

$$
\begin{aligned}
\delta_1 &= \frac{y_2 - \tan\alpha_1\, x_2}{\tan\alpha_1\,(x_1 - x_2) - y_1 + y_2} \\
\gamma_1 &= \frac{y_2 x_1 - y_1}{\tan\alpha_1\,(x_1 - x_2) - y_1 + y_2}.
\end{aligned}
\tag{78}
$$

The point where the lateral boundary geodesic and the discontinuity boundary intersect is a distance of $\gamma_1\sqrt{1+\tan^2\alpha_1} = \frac{\gamma_1}{|\cos\alpha_1|}$ from the starting point of the reference geodesic, to the right (or to the left if γ_1 is negative). If the angle $\alpha_1$is smaller than $\frac{\pi}{2}$, the inner side is to the right of the reference point. The value of $\cos\alpha_1$ is then positive, but the intersection point would be in the negative direction from the reference point. Equally, if the angle is greater than $\frac{\pi}{2}$, the inner side is to the left from the reference point, cosine is negative and the intersection point would be in the positive direction from the reference point. Therefore the difference in radius between the reference geodesic start point and the intersection point is

$$
\Delta R_1 = \frac{-\gamma_1}{\sqrt{\cos^2\theta_i + \sin^2\theta_i + m^2}\ \cos\alpha_1}. \tag{79}
$$

For the other side of the bend, simple linear algebra gives the intersection point

$$\delta_2 \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + (1-\delta_2) \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} L \\ 0 \end{bmatrix} + \gamma_2 \begin{bmatrix} 1 \\ \tan(\alpha_2) \end{bmatrix}. \tag{80}$$

The parameters δ_2 and γ_2 are solved

$$\begin{aligned} \delta_2 &= \frac{y_2 - \tan\alpha_1 \, x_2}{\tan\alpha_1 \, (x_1 - x_2) - y_1 + y_2} \\ \gamma_2 &= \frac{y_2 x_1 - y_1 - L\tan\alpha_2}{\tan\alpha_2 \, (x_1 - x_2) - y_1 + y_2} - L \, . \end{aligned} \tag{81}$$

The difference in radius between the reference geodesic end point and the intersection point on the side of the geodesic end point is therefore

$$\Delta R_2 = \frac{\gamma_2}{\sqrt{\cos^2\theta_i + \sin^2\theta_i + m^2}\,\cos\alpha_1}. \tag{82}$$

4.5.4 LATERAL BOUNDARY ALONG THE CIRCULAR DIRECTRIX CURVE

In this section a lateral boundary of a conical bend with the lateral boundary following a circular directrix curve is studied. The designer would typically want the lateral boundary to be an arc of a circle, and therefore go along the circular directrix curve. See 4.6.2 for specific examples. If both of the lateral boundaries are of the circular type, the distance between the lateral boundaries, or the width of the bend is easy to define.

However, it is not obvious to place the boundary. The aim is to construct a boundary which is not too far away from its endpoints. The total length of the lateral boundary including the plate extension could be minimized. It would be formulated as follows. The length of the circle arc

$$L_{curve} = (2\,\theta_0)R. \tag{83}$$

The distance of the polygon point to the end of the circle arc is

$$\begin{aligned} L_1 &= \left\| \boldsymbol{s}_{-\theta_0}(R) - \boldsymbol{x}_{E_1} \right\| \\ L_2 &= \left\| \boldsymbol{s}_{\theta_0}(R) - \boldsymbol{x}_{E_2} \right\| . \end{aligned} \tag{84}$$

Therefore the total length can be formulated the following way

$$L_{\text{total}} = L_{curve} + L_1 + L_2. \tag{85}$$

This formulation leads to a problem which is difficult to solve analytically and does not always have a unique solution. For instance, in the case of two edges parallel to each other, any directrix curve between the points would have the same length. Minimizing the squared sum of distances from the corner point of the polygon to the end point of the lateral boundary would solve the problem. This approach would place the lateral boundary so, that it would be equally far from the corner points on either side, if the bend was cylindrical. Squared distance minimizing, circular directrix curve following lateral boundary of a conical bend is calculated by minimizing the squared length

$$\begin{aligned}\sum \mathrm{L}^2{}_{\mathrm{i}} = &\left(\mathrm{R}\cos-\theta_0 - \mathrm{x}_{\mathrm{E}_1}^{\mathrm{r}_0}\right)^2 + \left(\mathrm{R}\sin-\theta_0 - \mathrm{x}_{\mathrm{E}_1}^{\mathrm{h}_0\times\mathrm{r}_0}\right)^2 + \left(\mathrm{mR} - \mathrm{x}_{\mathrm{E}_1}^{\mathrm{h}_0}\right)^2 \\ &+\left(\mathrm{R}\cos\theta_0 - \mathrm{x}_{\mathrm{E}_2}^{\mathrm{r}_0}\right)^2 + \left(\mathrm{R}\sin\theta_0 - \mathrm{x}_{\mathrm{E}_2}^{\mathrm{h}_0\times\mathrm{r}_0}\right)^2 + \left(\mathrm{mR} - \mathrm{x}_{\mathrm{E}_2}^{\mathrm{h}_0}\right)^2.\end{aligned} \tag{86}$$

By calculating the derivative of the Equation above and solving its root for R, we get

$$R = \frac{m x_{E_1}^{h_0} + m x_{E_2}^{h_0} + x_{E_1}^{r_0}\cos\theta_0 + x_{E_2}^{r_0}\cos\theta_0 - x_{E_1}^{h_0\times r_0}\sin\theta_0 + x_{E_2}^{h_0\times r_0}\sin\theta_0}{2+2m^2}. \tag{87}$$

However, in the specific use case to be presented in Section 4.6.2.1, the lateral boundary is not placed equally far from both polygon points, but so that the boundary is as close as possible to the other polygon point that minimizes the length of the curved section. Therefore, if R is the outer lateral boundary, R is the minimum of the distances of the polygon points from the equivalent discontinuity boundaries. If the inner boundary is derived, R is the maximum of the projections from the polygon point to the equivalent discontinuity boundary.

$$R = \begin{cases} \min\limits_i \left(\mathrm{proj}_{\bar{\boldsymbol{\delta}}_i}\left(\bar{\boldsymbol{x}}_j\right)\right), & \text{if } R \text{ is the radius of the outer lateral boundary} \\ \max\limits_i \left(\mathrm{proj}_{\bar{\boldsymbol{\delta}}_i}\left(\bar{\boldsymbol{x}}_j\right)\right), & \text{if } R \text{ is the radius of the inner lateral boundary.} \end{cases}$$

The major drawback of this formulation is that the area of the bend can become negative. As a result, if the lateral boundary is chosen using this method, it first has to be made sure that the projections of the polygon points on both sides have a common positive area.

4.6 AUTOMATICALLY SELECTING THE EDGES TO CONNECT

4.6.1 MOTIVATION

When the software user is modeling bent plate structures, often she wants to connect several edges of one polygonal plate to several edges of another polygonal plate. The polygons may have quite many edges and thus the software cannot assume that the user picks all the edges that he wants to connect from both plates, and thereafter manually defines the parameters of the bend. Therefore, an automated way of determining the topology of the bend, i.e. the selected edges to be connected is needed. And even if there are only a few edges to connect, it is reasonable to automate the selection of edges for the most common use cases.

In this section, the typical use cases and users' responses to them are identified, and based on them, common desirable properties of automatically selected bends are recognized. Then a multiple criteria decision making model is made based on the desirable properties of bend edge selection. Using the decision making model designed, automated selection of bend edges is implemented. The aim is to develop an algorithm that automatically selects the edges for creating the bend and creates an initial bend, always making sure, that the selected edges and parameters produce a feasible bend, which helps the user save time and effort.

4.6.2 TYPICAL USE CASES AND EXPECTED RESULTS

Based on discussion with Tekla business unit representatives, as well as the documents and presentation they provided, I have tried to identify the most typical and most important use cases where bent steel plate structures are used. This section presents those result. Based on these use cases a decision model is discussed in the following chapters.

4.6.2.1 CIRCULAR PARALLEL BENDS WITH A SMALL RADIUS

The most typical use case is where the user wants to create a folded plate structure, which consists of elemental plates and parallel bends with a small radius connecting the elemental plates. The structure should be such that it can be constructed of one plate by folding it possibly several times.

The user would place the elemental plates so, that the edges to be connected would be close to each other. The user would also expect that the lateral boundaries of the bent section are perpendicular to the bent section, so that the "coinciding" area of the edges

forms the bent section. If the corners of the elemental plates are close to each other, the user would expect the lateral boundary of the curved area to connect the corners.

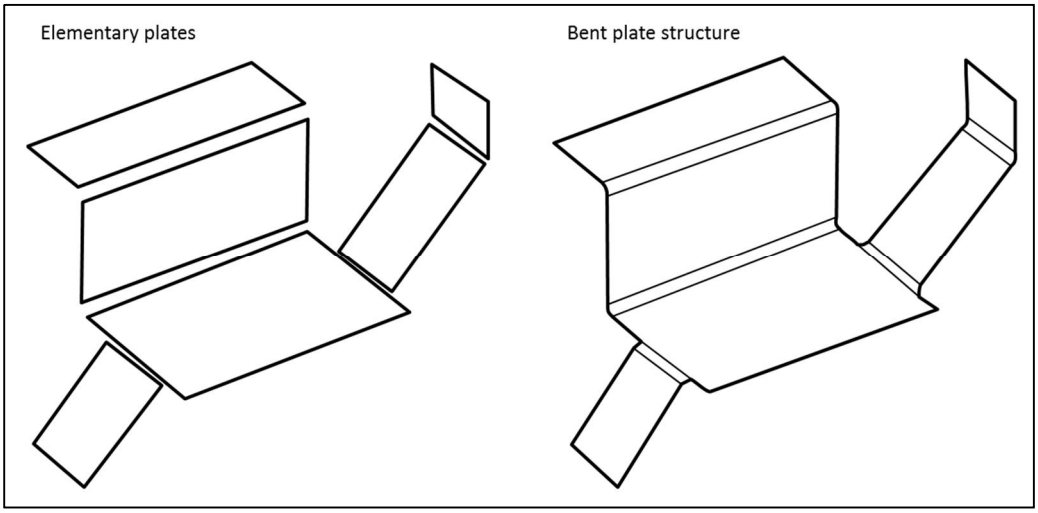


Figure 24. Connecting several elemental plates using parallel bends with a small radius.

4.6.2.2 LOFTED BENDS

One of the most typical bent plate structures are so called lofted bends that are normally used to build lofted transitions such as a part that connects a round pipe to a rectangular pipe. The user would model the flat edges as plates and expect that the computer automatically creates a lofted bend according to them.

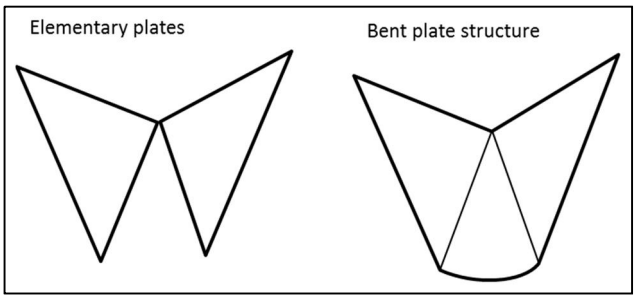


Figure 25. Connecting elemental plates to create a lofted bend

4.6.2.3 CURVED METAL PANELS

One of the main purposes of using bent plates is to create structures with curved metal panels. Normally the panels are cylindrical or conical (with a circular base). The user would model straight panels whose “facing” edges would be connected by a curved section.

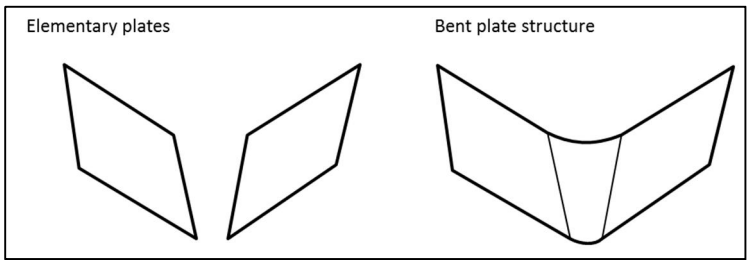


Figure 26. Curved metal panels with a conical bend

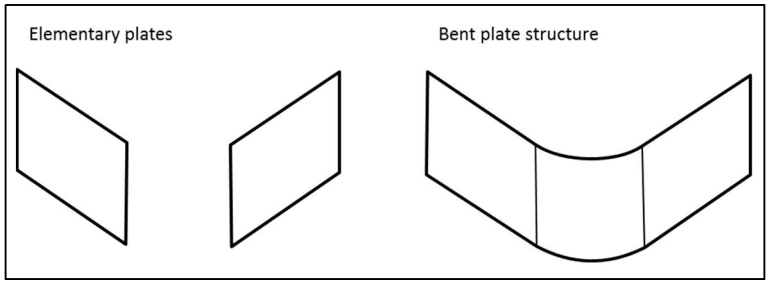


Figure 27. Curved metal panels with a parallel bend

If there are facing edges at a really low angle, the user is probably not willing to connect those by adding a significant plate extension and getting only a little wider bent section. This is illustrated in Figure 28.

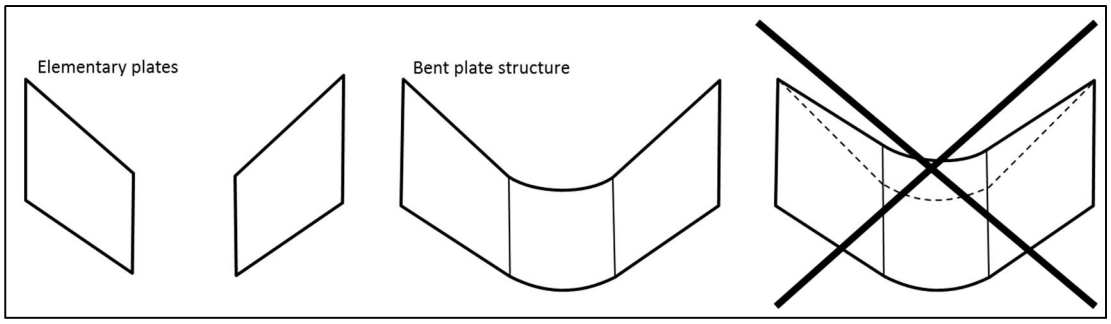


Figure 28. Facing edges at a low angle should not be connected

4.6.3 DECISION CRITERIA

In all the use cases presented in 4.6.2, the curved sections are either circular cylindrical or circular conical. Thus it can be assumed that the automatically added bends are either circular cylindrical or right circular conical.

In Section 4.6.2.1, the edges to be connected are selected in such a way, that the distance between the connected edges is as short as possible. Therefore the user wants to minimize the distance between the edges. In Section 4.6.2.2, the "distance" is also minimized. With a conical bend however, it is not clear how to define the "distance". Essentially the user wants to minimize the length of the bend, but still maximize the width of the bend, so that it spans a wide enough area between both of the elemental plate. This assumption is also valid in Section 4.6.2.3. Only the edges closest to each other are chosen, and if the edges are not parallel, the conical bend minimizing the area of the bend is chosen. Also, if there are edges facing each other at a very small angle, and thus adding considerably to the area, but not adding to the width, the edges are not selected.

Based on the observations above, the connected edges need to be selected so that:

- A conical or a cylindrical section can be added between all of the edges.
- The bent section is as wide as possible
- The average length of the bend section needs to be as small as possible
- The area of the plate extension needs to be as small as possible
- The edges need to be selected in such a way, that no holes appear in the curved section. The edges on either side have to form one continuous string of edges, or an open polygon.

4.6.4 CURVED SECTION WIDTH

There is no unambiguous way to define the width of a bend. The lateral boundaries are essentially two curves, and one can measure the distance between curves in many different ways, such as the average distance between all points, or the minimum distance between the points on one side and the other. For the sake of simplicity, only lateral boundaries following the circular directrix curve as presented in Section 4.5.4 are studied. Because the distance from one lateral boundary to the other along the generator line is a constant in this case, the width of the bend, w is defined as that distance. In the case of a cylinder, the curved section width would simply be the height of the cylinder. The case of a conical bend is

$$w = (R_2 - R_1)m\,. \tag{88}$$

What still needs to be decided is which directrix curve the lateral boundary would go. This will be decided with the formulation of the automated topology selection optimization problem later in 4.6.8.

4.6.5 MULTI-OBJECTIVE OPTIMIZATION PROBLEM SETTING

Based on the decision criteria given in 4.6.3, an optimization problem maximizing the bend width and minimizing the bend length is introduced in this section. The bend parameters and variables determining the topology of the bend – i.e. which edges are chosen to be connected – form the decision variables of the optimization problem. The problem of finding the best bend topology is a multi-objective optimization problem, because there are multiple objectives. The two objectives are the width w that should be maximized, and the average length L_{avg} that should be minimized. The variables of the optimization problem consist of

- Binary variables defining, whether an edge belongs to the bend or not,
- Parameters of the surface of the bend
- Lateral boundaries of the bend.

To fit a circular cylindrical or conical bend between two sets of edges, the intersection line between the planes of the elemental plates should not intersect the edges that are to be connected. So, before starting to find the set of optimal selected edges, the set of viable edges, i.e. the edges that do not intersect with the intersection line of the planes, is calculated. That can be done with an algorithm similar to that in Appendix 1. Let the binary vector marking the indices of feasible edges on side 1 be denoted E_1 and on side 2 E_2.

Also, when finding the viable edges, one has to make sure that on both sides the edges' normal (elemental plate border segment normal) are either pointing towards or away from the intersection line. Thus for both polygons, all such sets of consecutive edges are specified, where all the normal vectors of the edges are pointing either towards or away from the intersection line, and they are not crossing the intersection line. A pair with one set from both sides pointing the same direction form the feasible edges E_1 and E_2. All pairs need to be considered.

Because the problem presented here is a multi-objective optimization problem, there is no single objective function to minimize. Therefore there is usually no with a single

optimal solution. Marler & Arora (2004) describe the solution to a multi-objective optimization problem as "more of a concept than a definition." Therefore, concepts such as Pareto-optimality are used to find the set of "optimal" solutions. The idea of Pareto-optimality is to find the set of such solutions for which there exist no other points that would improve an objective without worsening the other.

Efficiency is a concept very similar to Pareto-optimality, and in many practical applications such as this, it is equivalent. Efficiency is defined as follows:

> *A point, $x^* \in X$, is efficient iff there does not exist another point, $x \in X$, such that $F(x) \leq F(x^*)$ with at least one $F_i(x) < F_i(x^*)$. Otherwise, x^* is inefficient. (Marler & Arora, 2004)*

The efficient frontier is the set of efficient points. There are two ways of presenting the efficient frontier in this case, either as the maximum width w for a given value of average length L_{avg}, or as the minimum average length for a given width. It is simpler to first find the minimum and maximum width of the bend, and then find the optimum areas for widths between those values.

Another way to treat multi-objective optimization problems, other than finding the set of efficient or Pareto optimal solutions, is scalarization which bombines the different objective functions to a scalar. Maybe the simplest example of scalarization is to calculate a weighted sum of the different objectives and minimize it.

Independent of the approach chosen, a single optimal solution is needed. Therefore it has to be understood, what sort of tradeoff the user is willing to accept between the variables L_{avg} and w. The tradeoff cannot be modeled using a simple weighting of the values in absolute terms. This means that, the plates to be connected can be relatively far away, and then the area would be relatively high compared to the width of the bend, and the optimal solution would be that no bends are created. It would be much more sensible to try to evaluate the tradeoff using width and area in relative terms.

To find a reference value for the length of the bend, let us first find the bend that connects one edge on both sides with the minimum length. The bend with the minimum length is a Pareto optimal solution, because its width cannot be expanded without increasing the area of the bend, and its area cannot be diminished without diminishing its width. The minimum length given by the solution is then used to evaluate how far away the plates are in terms of bend length.

4.6.6 FINDING THE BEND WITH THE MINIMUM AVERAGE LENGTH

One way to find a reference measure for the distance between the plates, is to calculate the minimum length of a bend between two edges, one on either side. There are at least a few ways of solving the minimization problem. In this thesis, four different ways to find the minimum distance are identified.

The simplest way to solve this is by going through all $n_1 n_2$ possible combinations of values of $\boldsymbol{\epsilon}_1$ and $\boldsymbol{\epsilon}_2$. For each combination, the plate extension minimizing linear programming problem would be solved as given in (37). Based on the parameters of the solution, bend average length would be calculated using either area and width of the bend, or by calculating the arithmetic mean of the lengths of both lateral boundaries, as given in (83) – (87).

This approach is straightforward, easy to implement and always finds the global optimal solution. However it is extremely slow, since $n_1 n_2$ linear optimization problems need to be solved in order to find the solution, where n_1 and n_2 are the number of edges on side 1 and 2 respectively. This method can be improved by not calculating the distance for all pairs of edges. For instance, one could measure arithmetic distance between the points, which is considerably faster, and then utilize the fact that the bend length is always equal or greater than the arithmetic distance.

A third way to find the bend with the minimum average length is to formulate a nonlinear multivariate optimization problem with the average length represented either as the quotient of the bend area and its width, or as the arithmetic mean of the length of the lateral boundaries. To solve the area and width, the following optimization problem needs to be solved

$$\begin{aligned} &\min_{\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2} \frac{A(\epsilon_1, \epsilon_2)}{w(\epsilon_1, \epsilon_2)} \\ &\boldsymbol{E}_1 \boldsymbol{\epsilon}_1 = 1 \\ &\boldsymbol{E}_2 \boldsymbol{\epsilon}_2 = 1\,, \end{aligned} \tag{89}$$

where $\boldsymbol{\epsilon}_1$ and $\boldsymbol{\epsilon}_2$ are binary vectors determining which edges are connected. The selected edges have to belong to the set of viable edges, and only one selected edge on either side is permitted, which gives the equality constraints. However, this formulation is a highly nonlinear and complicated optimization problem.

This sort of problems are called mixed integer non-linear programming problems. According to Costa & Oliveira (2001), "These problems, due to their combinatorial nature, are considered difficult problems." Especially for problems with non-convexities, such as the problem presented here, it is difficult to find the global optimum in a reasonable time. Also, because this optimization problem is only an initial sub-problem of the whole multi-objective optimization problem, such an approach can easily be computationally too complex.

The fourth way to solve the minimum bend length selection problem is to simplify the problem so that it can be formulated as a mixed integer linear programming (MILP) problem. The aim is to define the length of the bend as a linear function of the decision variables. This method requires many simplifications and assumes a different location of the lateral boundary, but it solves the global optimum of a sufficiently exact approximation of the length. Mixed integer programming is a well-known modeling technique and powerful solvers and algorithms for solving MILP problems exist.

MILP problems are NP-hard. This means that it is highly unlikely that there would be a polynomial time algorithm for solving the problem. In practice, the computational complexity and time required for solving the problem increases significantly as the number of binary variables increases. MILP problems that may seem simple, can be difficult to solve using any existing solvers. (Bertsimas & Tsitsiklis, 1997; Appa et al., 2006)

Therefore it is important to verify and test the computational efficiency of the MILP-approach and if necessary, find ways to formulate the problem in such a way that it can be solved effectively and try to cut out unnecessary binary variables. For instance, using Euclidian distance, one could narrow down the set of possible edges. Nonetheless, a MILP formulation of the problem is presented and implemented in this thesis due to its robustness and the fact that it finds the global optimum for the approximation.

4.6.7 THE MILP FORMULATION OF THE MINIMUM LENGTH BEND PROBLEM.

In this section, a MILP formulation for the minimum length bend optimization problem is presented. Because the objective of a MILP problem needs to be a linear function of the decision variables, a way to formulate the length of the bend as a linear function of the decision variables is presented. The idea is to calculate the average length of the bend as the sum of the average length of the curved section along the directrix curves and the

average length of the plate extension. To express the average length as a linear function of the decision variables, certain simplifications on the geometry need to be assumed:

1. The lateral boundary follows a directrix curve.
2. The lateral boundary end points are equally far on x-axis (the axis along the discontinuity boundary) from the corresponding polygon vertices on both sides of the bend.
3. The plate extension lateral edge is assumed to be perpendicular to the intersection line of the planes, as is done in Sections 4.1.7 and 4.1.5.
4. The lateral boundary is not assumed to be continuous or differentiable.

The geometry of the situation is depicted in Figure 29. As can be seen, this formulation may not be sufficient to define the lateral boundary of a bend, but it is certainly enough to estimate the average length of the bend. Let the average length of the bend be the sum of the average length of the curved section and the average length of the plate extension.

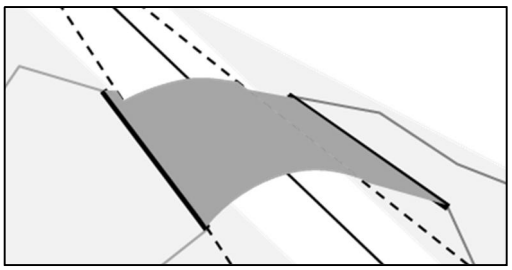


Figure 29. Bend whose length can be estimated linearly.

The items 1 and 2 above are there to make sure, that the average length of the curved section can be calculated as a linear function of the decision parameters x_s, β, $\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2$. The item 3 assures, that the plate extension length function is linear. Let us define the total average length as the sum of the average lengths of the curved section and the plate extensions

$$L_{\text{avg}} = L_{\text{avg,curve}} + L_{\text{avg,1}} + L_{\text{avg,2}} + L_{\text{shift}}. \tag{90}$$

The first term in this equation is the average length of the curved section. The second and the third terms are the average lengths of the plate extension. The final term represents how much the curved section lateral edge end point is shifted from the plate extension

corner point along the discontinuity boundary. Without this term, the distance would be one dimensional, and the length would only describe the selected bend average distance from the plane intersection line.

Let us first formulate the average curved section length as a linear function of the decision variables. The length of a circular directrix curve of a conical bend at radius R is simply $2\theta_0 R$. The azimuth angle θ_0 is

$$\theta_0 = \tan^{-1}\left(\sqrt{\frac{\sin^2\alpha + \beta^2}{\cos^2\alpha}}\right), \alpha \leq \frac{\pi}{2}. \tag{91}$$

The average radius is

$$R_{avg} = \frac{y_{avg}}{\tan(\theta_0)} = y_{avg}\sqrt{\frac{\cos^2\alpha}{\sin^2\alpha + \beta^2}}. \tag{92}$$

The length of the directrix curve is a non-linear function of the variable β. However in all reasonable use cases, β is relatively small, less than 0.5 in absolute terms. Also, it is known that $\theta_0 = \alpha$ when $\alpha \to \frac{\pi}{2}$. When $\alpha \to 0$, the cone angle is $\theta_0 = \tan^{-1}|\beta|$. When the slope β is small, the angle can then be approximated as $|\beta|$. The radius is also non-linear. The exact value of the length is

$$L_{avg,curve} = 2\theta_0 R = 2y_{avg}\sqrt{\frac{\cos^2\alpha}{\sin^2\alpha + \beta^2}}\tan^{-1}\left(\sqrt{\frac{\sin^2\alpha + \beta^2}{\cos^2\alpha}}\right). \tag{93}$$

When the angle β is within a reasonable range, or under 0.5, the length is 2 y_{avg} when the angle $\alpha = 0$. If the angle $\alpha \to \frac{\pi}{2}$, the length approaches zero. The exact value for the length is drawn in Figure 30.

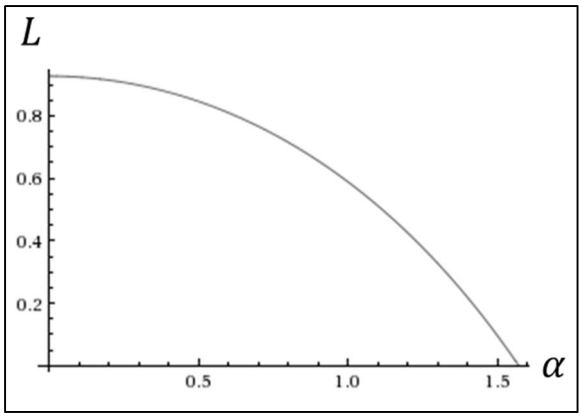


Figure 30. The relative average length of a conical curved section as a function of the variable β

Let us approximate the length of the bend with a function independent of β. The variable β is chosen a constant value so that the error term stays small within a reasonable range. By testing different values of β_0, the approximation in (94) is selected. The total error of the approximation with $y_{avg} = 1$ and different values of β is shown in Figure 31.

$$L_{avg,curve} \approx 2y_{avg}\sqrt{\frac{\cos^2\alpha}{\sin^2\alpha+\beta_0^2}}\tan^{-1}\left(\sqrt{\frac{\sin^2\alpha+\beta_0^2}{\cos^2\alpha}}\right) \qquad (94)$$

$$\beta_0 = 0.3\,.$$

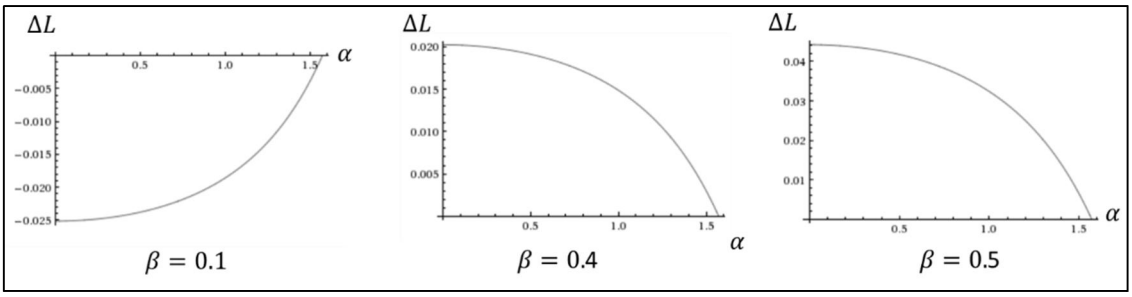


Figure 31. Error of the curved section linear approximation as a function of the angle between planes, for different values of the variable β.

As can be seen, the decision variable β has a relatively small effect on the approximation error and that within the reasonable range, the error remains small. Thus it is justified to approximate in the way done above. The term y_{avg} is the average distance from the discontinuity edge to the intersection line

$$y_{avg} = \frac{(\beta x_k^1 - \beta x_s) + (\beta x_l^2 - \beta x_s) + (\beta x_{k+1}^1 - \beta x_s) + (\beta x_{l+1}^2 - \beta x_s)}{4}. \quad (95)$$

Thus the linear approximation for the length of the curved section is

$$L_{avg,curve} \approx \left(\beta \frac{x_k^1 + x_l^2 + x_{k+1}^1 + x_{l+1}^2}{2} - 2x_\beta\right) \sqrt{\frac{\cos^2 \alpha}{\sin^2 \alpha + \beta_0^2}} \tan^{-1}\left(\sqrt{\frac{\sin^2 \alpha + \beta_0^2}{\cos^2 \alpha}}\right), \quad (96)$$

where $x_\beta = \beta\, x_s$. The length of the plate extensions can be calculated similarly as the plate extension areas in Section 4.1.4

$$\begin{aligned} L_{avg,1} &= ½(y_{k+1}^1 + y_k^1) - ½\beta(x_{k+1}^1 + x_k^1) + x_\beta \\ L_{avg,2} &= ½(y_{l+1}^2 + y_l^2) - ½\beta(x_{l+1}^2 + x_l^2) + x_\beta\,. \end{aligned} \quad (97)$$

Finally, let the shift be defined as the sum of the differences of the corner points' x-coordinate value on the outer and the inner side

$$L_{\text{shift}} = |x_k^1 - x_l^2| + |x_{k+1}^1 - x_{l+1}^2|\,. \quad (98)$$

Based on these results, a MILP model is created that minimizes the linear approximation of the length of the bend. The decision variables consist of the bend parameters: the discontinuity boundary line slope β, the location of the cone apex x_s, and the indices of the selected edge corner points $\boldsymbol{\epsilon}$. The minimum length bend MILP optimization problem is formulated as follows

$$\min \quad \left((\Psi_\alpha - 1)\left(\frac{\sum y_{\beta 1}}{2} + \frac{\sum y_{\beta 2}}{2}\right) + \sigma_1 + \sigma_2 + \frac{1}{2}\boldsymbol{y}^1(\boldsymbol{\epsilon}_1^1 + \boldsymbol{\epsilon}_2^1) + \frac{1}{2}\boldsymbol{y}^2(\boldsymbol{\epsilon}_1^2 + \boldsymbol{\epsilon}_2^2)\right) \tag{99}$$

$$\text{s.t.} \quad \beta x_k^i - x_\beta \le y_k^i + \left(1 - \boldsymbol{\epsilon}_1^i(k) - \boldsymbol{\epsilon}_2^i(k)\right) M$$

$$y_{\beta i}(k) \ge 0$$

$$y_{\beta i}(k) \ge \beta x_k^i - x_\beta - \left(1 - \boldsymbol{\epsilon}_1^i(k) - \boldsymbol{\epsilon}_2^i(k)\right) M$$

$$y_{\beta i}(k) \le \beta x_k^i - x_\beta + \left(1 - \boldsymbol{\epsilon}_1^i(k) - \boldsymbol{\epsilon}_2^i(k)\right) M$$

$$y_{\beta i}(k) \ge -\left(\boldsymbol{\epsilon}_1^i(k) + \boldsymbol{\epsilon}_2^i(k)\right) M$$

$$y_{\beta i}(k) \le \left(\boldsymbol{\epsilon}_1^i(k) + \boldsymbol{\epsilon}_2^i(k)\right) M$$

$$\mathbf{1}^T\boldsymbol{\epsilon}_1^i = 1,\ \mathbf{1}^T\boldsymbol{\epsilon}_2^i = 1$$

$$\boldsymbol{\epsilon}_1^i(k) + \boldsymbol{\epsilon}_2^i(k) \le 1$$

$$-1 \le \boldsymbol{C^T}\boldsymbol{\epsilon}_1^i - \boldsymbol{C^T}\boldsymbol{\epsilon}_2^i \le 1$$

$$\sigma_j \ge 0,\ \sigma_j \ge \boldsymbol{\epsilon}_j^1\boldsymbol{x}^1 - \boldsymbol{\epsilon}_j^2\boldsymbol{x}^2,\ \sigma_1 \ge \boldsymbol{\epsilon}_j^2\boldsymbol{x}^2 - \boldsymbol{\epsilon}_j^1\boldsymbol{x}^1$$

$$\boldsymbol{\epsilon}_1^i \in \text{bin}^{n_i}, \boldsymbol{\epsilon}_2^i \in \text{bin}^{n_i},$$

where $\mathbf{1}^T = [1 \quad 1 \quad \dots \quad 1]$, $\boldsymbol{C^T} = [1 \quad 2 \quad 3 \quad \dots \quad n]$, $i \in \{1,2\}$, $j \in \{1,2\}$ and $\Psi_\alpha = \sqrt{\frac{\cos^2\alpha}{\sin^2\alpha+\beta_0^2}}\tan^{-1}\left(\sqrt{\frac{\sin^2\alpha+\beta_0^2}{\cos^2\alpha}}\right)$. There are two sides in a bend. The index i determines the side of the bend. In both sides there are two selected edges. The index j determines, which of the two selected edges is in question. In linear optimization, the decision variables should not be multiplied in the objective function or the constraints. However, because the apex location x_s is always multiplied by the slope variable β, a new decision variable $x_\beta = x_s\beta$ is introduced, and the value of the variable x_s can be calculated after solving the problem.

In the optimization model, many changes are made to express the problem as a MILP problem. The constant Ψ_α is introduced, since it is only dependent on α and β_0, both constants. New variables $y_{\beta 1}$ and $y_{\beta 2}$ are introduced. Their values should equal to $\beta x_k^1 -$

x_β and $\beta x_l^2 - x_\beta$ at the selected points respectively. The so called Big-M method is used to set the values of the variables $y_{\beta 1}$ and $y_{\beta 2}$ to the value given above only for the selected points, and to zero elsewhere. To calculate the shift, absolute value needs to be calculated. Because absolute value is not a linear function, it has been linearized by introducing new variables (σ_1 and σ_2) and constraints. For more on these methods, see Bertsimas & Tsitsiklis (1997).

The binary vector variables $\boldsymbol{\epsilon_1^1}$, $\boldsymbol{\epsilon_2^1}$, $\boldsymbol{\epsilon_1^2}$ and $\boldsymbol{\epsilon_2^2}$ are determining the indices of the polygon end points that are taken to calculation. On both sides there has to be two selected polygon end points. The points have to be identified, that is why two binary vectors with only one value 1 on each side is introduced, instead of just having one binary vector that has two non-zero variable values. Because the points have to be adjacent, the constraints $-1 \leq \boldsymbol{C^T\epsilon_1^1} - \boldsymbol{C^T\epsilon_2^1} \leq 1$ and $-1 \leq \boldsymbol{C^T\epsilon_1^2} - \boldsymbol{C^T\epsilon_2^2} \leq 1$ are introduced. The constraints $\boldsymbol{\epsilon_1^1}(k) - \boldsymbol{\epsilon_2^1}(k) \leq 1$ and $\boldsymbol{\epsilon_1^2}(l) - \boldsymbol{\epsilon_2^2}(l) \leq 1$ ensure that the selected points on each side are different.

This formulation is based on the assumption that the edge normals of the feasible set of edges are facing the plane intersection line. This is necessarily not the case, as mentioned before. The edge normals can also be pointing away from the intersection line on both sides. Then the azimuth angle used in calculating the length of the curved section would be the difference of π and the azimuth angle given in (91). The constant Ψ_α would then be calculated differently

$$\Psi_{\alpha_{long}} = \sqrt{\frac{\cos^2\alpha}{\sin^2\alpha+\beta_0^2}}\left(\pi - \tan^{-1}\left(\sqrt{\frac{\sin^2\alpha+\beta_0^2}{\cos^2\alpha}}\right)\right). \tag{100}$$

Also, if the edges are facing away from the plane intersection line, the constraints too need to be changed so that discontinuity boundaries are on the outside of the plates:

$$\begin{aligned} \beta x_k^1 - x_\beta &\geq y_k^1 - \left(1 - \boldsymbol{\epsilon_1^1}(k) - \boldsymbol{\epsilon_2^1}(k)\right)M \\ \beta x_l^2 - x_\beta &\geq y_l^2 - \left(1 - \boldsymbol{\epsilon_1^2}(l) - \boldsymbol{\epsilon_2^2}(l)\right)M\,. \end{aligned} \tag{101}$$

The plate extension length too needs to be calculated differently with edges facing away from the intersection line, taking into account the fact that now the discontinuity boundary is on the other side of the plate. Thus the objective needs to be rewritten

$$L_{avg} = \left(\left(\Psi_{\alpha_{long}} + 1 \right) \left(\frac{\sum y_{\beta 1}}{2} + \frac{\sum y_{\beta 2}}{2} \right) + \sigma_1 + \sigma_2 - \frac{1}{2} \boldsymbol{y}^1 (\boldsymbol{\epsilon}_1^1 + \boldsymbol{\epsilon}_2^1) - \frac{1}{2} \boldsymbol{y}^2 (\boldsymbol{\epsilon}_1^2 + \boldsymbol{\epsilon}_2^2) \right). \quad (102)$$

4.6.8 THE BEND TOPOLOGY MULTI-OBJECTIVE OPTIMIZATION PROBLEM FORMULATION

In this section, the multi objective optimization problem for finding the optimal topology of a bend is formulated as a MILP problem. The attempt in the multi-objective optimization problem is to understand what sort of tradeoff the user is willing to accept between the length and the width of the bend. As already discussed, the distance between the plates should not affect the results. Let the net length be defined as the difference between the length of the bend and the minimum length as solved in 4.6.6

$$\Delta L = L - L_{\min} . \quad (103)$$

Similarly, let the net width be defined as the addition of width compared to the bend with the minimum length

$$\Delta w = w - w_{L_{min}}. \quad (104)$$

Essentially we wish to assure that adding more width to the bend does not add too much to the bend length. Therefore the multi-objective optimization problem could be simplified to a single objective MILP-problem, where the width is maximized and the length is controlled based on the tradeoff. The constant ξ determines the threshold value how much the net length can be in relation to the net width.

The optimization problem is formulated as a MILP-problem

$$\begin{aligned} &\min -w \\ &s.t. \Delta L \leq \xi \Delta w . \end{aligned} \quad (105)$$

First, the width w needs to be written in a linear form using a similar linearization as in Section 4.6.7. This is trivial, because of the assumption in Section 4.6.6, which states that the lateral boundary end points should be equally far on the axis x. The width can then be

therefore calculated as the difference of the average x-values of the corner points, or simply as the average edge length projected to the intersection curve over both sides

$$w = \frac{|x_k^1 - x_{k+1}^1| + |x_l^2 - x_{l+1}^2|}{2}. \tag{106}$$

As in the MILP problem for finding the minimum length bend, binary variables are used to determine which polygon corner points on either side are chosen. In the minimum length bend problem, only two consecutive points are chosen, whereas in this problem, the attempt is to find a string of polygon points on either side. Therefore the binary vectors determining the selected corner points have to consist of either a string of consecutive ones and the rest zeroes, or a string of consecutive zeroes and the rest ones. A binary vector with one string of consecutive ones can be generated as follows

$$\begin{aligned} &C\boldsymbol{\epsilon} \leq \iota_{max} \\ &C\boldsymbol{\epsilon} \geq \iota_{min} - (1 - \epsilon_1)M \\ &\mathbf{1}^T\boldsymbol{\epsilon} = (1 + \iota_{max} - \iota_{min}) . \end{aligned} \tag{107}$$

To define a measurement for the bend length in the case of connecting two arbitrary length arrays of edges using a linear formulation is not completely straightforward. As with the minimum length bend problem, the shift has to be taken into account. Also, it needs to be decided whether the length is calculated in terms of the maximum distance or the average distance. There is no obvious pairing between the polygon corner points on one side and another.

We seek to define the length rather in terms of the absolute (maximum local) length than in terms of the average length. For example, consider the case depicted in Figure 32. If the bend on the dark gray area were to be augmented with the lighter gray area, the average length would increase only a little, but the absolute length would increase significantly. In this case, the addition would not be desirable, so the length should be measured based on the absolute length.

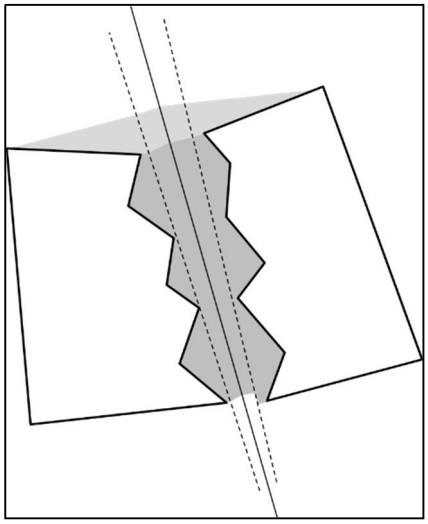


Figure 32. Augmenting plate extension causes a major increase in the absolute length, but only a minor increase in average length.

Therefore let the absolute length of the bend be measured using the following function

$$L_{span} = L_{\text{curve,max}} + L_{1,\text{max}} + L_{2,\text{ max}} + L_{shift}. \tag{108}$$

Let us call this function the length span. It is the sum of four terms: **1.** the maximum length of the curved section circular directrix curves, **2.** the maximum distance of the polygon corner point from the discontinuity boundary, on a line perpendicular to the intersection line, over all selected points in polygon 1, and **3.** the same maximum distance over the selected corner points of the second polygon. The fourth term is the shift along the x-axis as defined in Section 4.6.6.

The curved section maximum length can be calculated as

$$L_{\text{curve, max}} = 2y_{avg}\Psi_\alpha = \beta\left(\max_k\{(x_k^1 - x_s)\} + \max_k\{(x_k^1 - x_s)\}\right)\Psi_\alpha. \tag{109}$$

Using the same formulation as in (99), the curved section length can be expressed as a linear function

$$\begin{aligned} &L_{curve,max} = \Psi_\alpha\left(y_{\beta 1-max} + y_{\beta 2-max}\right) \\ &y_{\beta 1-max} \geq y_{\beta 1}(k) \forall\ k \in n_1 \\ &y_{\beta 1-max} \geq y_{\beta 2}(l) \forall\ l \in n_2\ . \end{aligned} \tag{110}$$

The maximum distance from the polygon corner points to the discontinuity boundary on either side also needs a new variable

$$\begin{aligned} L_{1,\max} &\geq y_k - \beta x_k^1 + x_\beta \; \forall k \in n_1 \\ L_{2,\max} &\geq y_k - \beta x_l^2 + x_\beta \; \forall l \; \in n_2. \end{aligned} \tag{111}$$

The shift is simply the sum of the differences of the extreme values of the variable x from side to side

$$L_{shift} = x_{max}^{shift} + x_{min}^{shift} = |x_{\text{max-x}}^1 - x_{\max}^2| + |x_{\text{max-x}}^1 - x_{\max}^2|. \tag{112}$$

The length span can therefore be expressed as

$$L_{span} = \Psi_\alpha\left(y_{\beta 1-max} + y_{\beta 2-max}\right) + L_{1,\max} + L_{2,\max}. \tag{113}$$

The net length span is the difference of the length span and the minimum length as calculated in the previous MILP problem

$$\begin{aligned} \Delta L &= L_{span} - L_{min} \\ &= \Psi_\alpha\left(y_{\beta 1-max} + y_{\beta 2-max}\right) + L_{1,\max} + L_{2,\max} - L_{min}. \end{aligned} \tag{114}$$

The linear optimization problem maximizing the total width can therefore be expressed as a MILP problem:

$$\min -\frac{\boldsymbol{x}_i^1\boldsymbol{\epsilon}_{max}^1 - \boldsymbol{x}_i^1\boldsymbol{\epsilon}_{min}^1}{2} - \frac{\boldsymbol{x}_i^2\boldsymbol{\epsilon}_{max}^2 - \boldsymbol{x}_i^2\boldsymbol{\epsilon}_{min}^2}{2} + \frac{1}{\mathrm{M}}\left(\sum\left(\epsilon^1(k)y_k^1 - y_{\beta 1}(k)\right) + \sum\left(\epsilon^2(l)y_l^1 - y_{\beta 2}(l)\right)\right)$$

$$s.t.$$

$$\boldsymbol{\epsilon}^i(k) - \boldsymbol{\epsilon}_{max}^i(k) \geq 0,\ \boldsymbol{\epsilon}^i(k) - \boldsymbol{\epsilon}_{min}^i(k) \geq 0\ \ \forall k \in n_i$$

$$\mathbf{1}^T\boldsymbol{\epsilon}_{max}^i = 1, \mathbf{1}^T\boldsymbol{\epsilon}_{min}^i = 1$$

$$\Psi_\alpha\left(y_{\beta 1-max} + y_{\beta 2-max}\right) + L_{1,\max} + L_{2,\max} + x_{max}^{shift} + x_{min}^{shift} - L_{min}$$

$$\leq \xi\left(\frac{\boldsymbol{x}_i^1\boldsymbol{\epsilon}_{max}^1 - \boldsymbol{x}_i^1\boldsymbol{\epsilon}_{min}^1}{2} + \frac{\boldsymbol{x}_i^2\boldsymbol{\epsilon}_{max}^2 - \boldsymbol{x}_i^2\boldsymbol{\epsilon}_{min}^2}{2} - w_{L_{\min}}\right)$$

$$x_{max}^{shift} \geq 0, \qquad x_{max}^{shift} \geq \boldsymbol{x}_i^1\boldsymbol{\epsilon}_{max}^1 - \boldsymbol{x}_i^2\boldsymbol{\epsilon}_{max}^2, \qquad x_{max}^{shift} \geq \boldsymbol{x}_i^2\boldsymbol{\epsilon}_{max}^2 - \boldsymbol{x}_i^1\boldsymbol{\epsilon}_{max}^1$$

$$x_{min}^{shift} \geq 0, \qquad x_{min}^{shift} \geq \boldsymbol{x}_i^1\boldsymbol{\epsilon}_{min}^1 - \boldsymbol{x}_i^2\boldsymbol{\epsilon}_{min}^2, \qquad x_{min}^{shift} \geq \boldsymbol{x}_i^2\boldsymbol{\epsilon}_{min}^2 - \boldsymbol{x}_i^1\boldsymbol{\epsilon}_{min}^1$$

$$\iota_{min}^1 - (1 - \boldsymbol{\epsilon}_1^1)M \leq \boldsymbol{C}(k)\boldsymbol{\epsilon}_1^i(k) \leq \iota_{max}^i\ \ \forall k \in n_i$$

$$\mathbf{1}^T\boldsymbol{\epsilon}_1^i = 1 + \iota_{max}^i - \iota_{min}^i$$

$$\iota_{max}^i - \iota_{min}^i \geq 1$$

$$\boldsymbol{\epsilon}^i(k) \geq 0\ \forall k \in n_i$$

$$\boldsymbol{\epsilon}^i(k) \geq \boldsymbol{\epsilon}_1^i(k) - (1 - \zeta_i)M\ \forall k \in n_i$$

$$\boldsymbol{\epsilon}^i(k) \leq \boldsymbol{\epsilon}_1^i(k) + (1 - \zeta_i)M\ \forall k \in n_i$$

$$\boldsymbol{\epsilon}^{\boldsymbol{i}}(k) \geq \left(1 - \boldsymbol{\epsilon}_1^i(k)\right) - \zeta_i M\ \forall k \in n_i$$

$$\boldsymbol{\epsilon}^{\boldsymbol{i}}(k) \leq \left(1 - \boldsymbol{\epsilon}_1^i(k)\right) + \zeta_i M\ \forall k \in n_i$$

$$y_{\beta i-max} \geq y_{\beta i}(k) \forall\ k \in n_i$$

$$L_{i,\max} \geq y_k^i - \beta x_k^i + x_\beta - \left(1 - \boldsymbol{\epsilon}^{\boldsymbol{i}}(k)\right)M\ \forall k \in n_i$$

$$\beta x_k^i - x_\beta \leq y_k^i + \left(1 - \boldsymbol{\epsilon}^{\boldsymbol{i}}(k)\right)M\ \forall k \in n_i$$

$$\beta x_k^i - x_\beta \geq 0 - \left(1 - \boldsymbol{\epsilon}^{\boldsymbol{i}}(k)\right)M\ \forall k \in n_i$$

$$y_{\beta i}(k) \geq 0, \qquad y_{\beta i}(k) \geq \beta x_k^i - x_\beta - \left(1 - \boldsymbol{\epsilon}^i(k)\right)M \quad \forall k \in n_i.$$

4.7 PROTOTYPE IMPLEMENTATION

To test the method for connecting flat elemental plates using the methods presented here, a prototype computer program is made. A 3D modeling suite is used to model the flat elemental plates. The 3D data will then be transferred to the prototype program using a plug-in of the 3D modeling suite specifically made for reading the 3D-data of the flat plates and transferring it to the prototype program. When the prototype program is finished, it generates a script file that creates the bent section 3D data in the 3D modeling suite.

The aim of the prototype is to verify, if the use cases presented in Section 4.6.2 can be handled by the automatic bend topology algorithm based on the MILP-formulation, and to study the computational complexity of the optimization problem. The prototype does not verify the geometric or topological unfoldability of the bent plate structures, but only finds the sets of edges that could be connected and generates the bends.

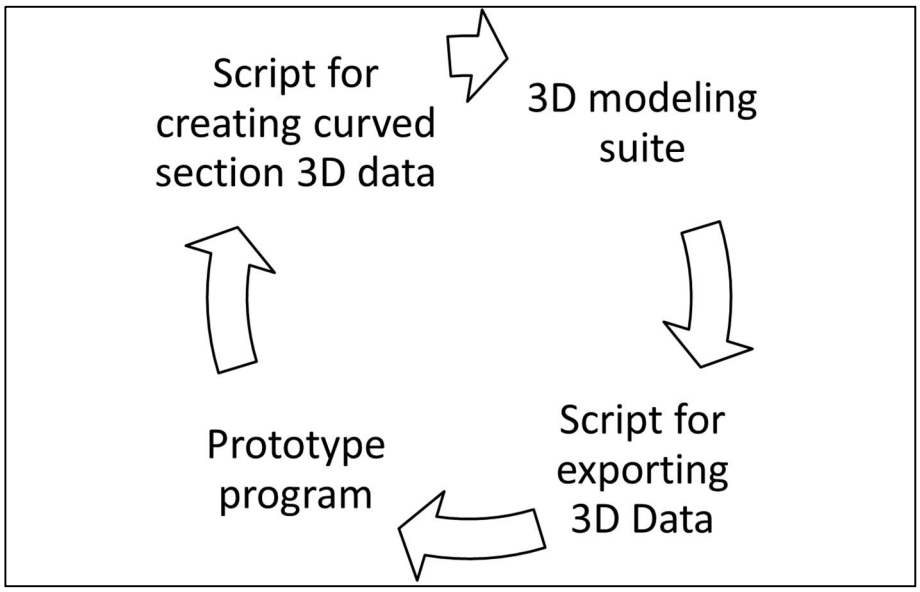


Figure 33. Data flow in the prototype implementation

The prototype program takes the geometry of two elemental plates as an input, interprets the 3D data provided by the modeling suite and calculates the minimum length bend using the MILP formulation presented in Section 4.6.7. Based on the minimum length calculated, the program selects the edges to be connected using the MILP formulation presented in Section 4.6.8. As a byproduct, other bend parameters are also calculated.

Then, if the bend slope is smaller than a given constant threshold, the conical curved section is replaced with a plate extension minimizing cylindrical curved section. Finally the lateral boundary is calculated according to the user input and the curved section and plate extension geometry is exported back to the 3D modeling suite using a generated script.

The prototype program is implemented using the C++ -programming language and a 64 bit Windows executable is compiled using Microsoft Visual Studio 2013 compiler. In addition to the standard C++ 11 libraries, a library for vector calculus is used. COIN-OR Symphony 5.5.0 is used for calculating MILP problems. The 3D modeling suite used is Trimble Sketchup, and the plug-in made for Trimble Sketchup is programmed using the Ruby scripting language.

5 COMPUTATIONAL RESULTS

The algorithm for finding the bend topology is tested using the prototype program and six different sets of elemental plates to be connected. The elemental plates in each use case are presented in Figure 34. The test case (a) has several simple planes to be connected to form a typical folded plate structure. This test case demonstrates the most typical use case where folded plates are used.

All other test cases only have two elemental plates that are then connected with either a conical or a cylindrical bend. The test case (b) is added to demonstrate how the developed algorithm could help in creating a lofted bend, that is used connect a round and a rectangular pipe. Test case (c) depicts a situation where a bent plate structure is created by ripping a plate before bending.

Test case (d) is used as a test piece to see what sort of results the algorithm gives when the input elemental plates are of a more irregular shape. The case (e) on the other hand is there to demonstrate the tradeoff between bend width and bend length. Finally, test case (f) is there to test the calculation time, when the number of edges is especially large. For each use case, the calculation time of both the minimum length calculations and the final solving of the optimization model is retrieved.

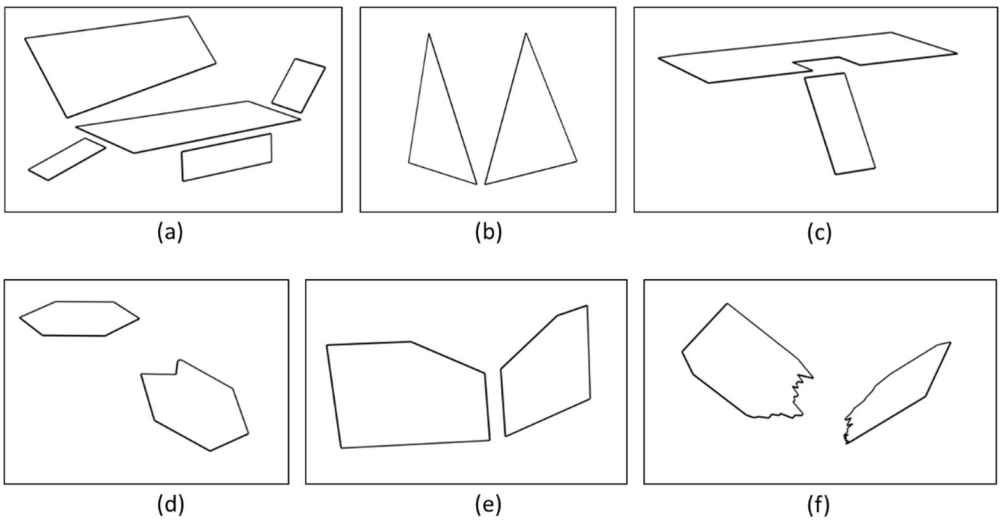


Figure 34. Elemental plates in use cases tested with the prototype program

Finding minimum length bend						
Test case	**a***	**b**	**c**	**d**	**e**	**f**
Time /ms	74 + 189	929 + 88	1813 + 107	1190 + 1950	1146 + 950	47245 + 2241
Number of all variables	16 + 16	19 + 16	34 + 16	22 + 34	19 + 22	121 + 34
Number of integer variables	8 + 8	10 + 8	20 + 8	12 + 20	10 + 12	78 + 20
Number of constraints	36 + 36	42 + 36	72 + 36	48 + 72	42 + 48	246 + 72
Finding optimum bend						
Test case	**a***	**b**	**c**	**d**	**e**	**f**
Time /ms	117	122	102	114	332	6972
Number of all variables	30	30	30	54	38	170
Number of integer variables	16	16	16	22	18	51
Number of constraints	63	63	63	135	87	483
* The test case **a** has several bends. The large conical bend results are reported here.						

Table 1. Number of variables and computing times of test cases.

A 64 bit binary of the prototype program was run with the test cases presented above. The test computer was running a Windows 7 Enterprise operating system and had an Intel Xeon E3-1270 central processing unit with 8 cores and 16 GB of memory. However, only three of the CPU cores were used for solving the MILP problem.

The computing times to calculate the minimum length bend and the optimum bend topology as well as the numbers of variables in each test case are given in Table 1. In the minimum length bend problem, two values are reported for all variables. This is because the minimum length bend problem needs to be solved for two different sets of edges – the edges facing the intersection line, and the edges facing away from the intersection line.

The most spectacular feature of the calculation times is that finding the minimum length bend takes a considerably longer time than solving the seemingly more complicated multi-objective optimization problem of finding the bend topology. The reason most probably lies in the formulation. According to Bertsimas & Tsitsiklis (1997), "Extensive computational experience suggests that the choice of a formulation is crucial [in integer programming]." Also, a great majority of the time spent in calculating the minimum length bend goes to finding the integer solution after having found the solution of the linear relaxation of the problem. All this evidence suggests that the formulation of the minimum length bend problem should be reformulated so that the optimum of the linear relaxation of the problem is closer to the optimum of the full MILP problem.

The problem with the formulation of the minimum length bend problem presented in 4.6.7 is that the binary variable vectors $\boldsymbol{\epsilon_1^1}$, $\boldsymbol{\epsilon_1^2}$, $\boldsymbol{\epsilon_2^1}$ and $\boldsymbol{\epsilon_2^2}$, determining which corner points are chosen on each side, can get values other than 1 or 0 as a result of the linear relaxation of

the optimization problem. However, this is not the case in the formulation of the optimum bend MILP problem presented in 4.6.8. Because of the way the index numbers are used to restrict the selected edges and to make sure the selected edges are consecutive, already solving the linear programming relaxation of the problem ensures that the binary variables indicating the selected edges are either 1 or 0.

In the bend topology optimization MILP problem, neither the number of all variables, number of integer variables nor the number of constraints seem to fully explain the computational time required. Especially the case (e) that seems quite simple, takes a considerably longer time to solve than the case (d) that has more edges, decision variables, constraints and looks more complicated. However the results of the case (f) suggest that the overall complexity of the problem adds to the calculation time of the algorithm.

The test cases with the bent sections calculated are presented in Figure 35. The results are as expected and cover the typical use cases given in Section 4.6.2. The aim of the development of the algorithm presented in this thesis, was to create a method that saves the user the time of not having to select the edges and determining all parameters manually to get visible results. The calculation time, even with the most complex test cases was relatively low and most probably a lot less than what the user would need to create the same bend inputting the parameters manually. However, before using the algorithm in any software product, the minimum length bend optimization problem would have to be reformulated in such a way, that it requires less computation time. This could be achieved by using a similar approach to restrict the indices of the selected edges, the same way as was done in the multi objective optimization model.

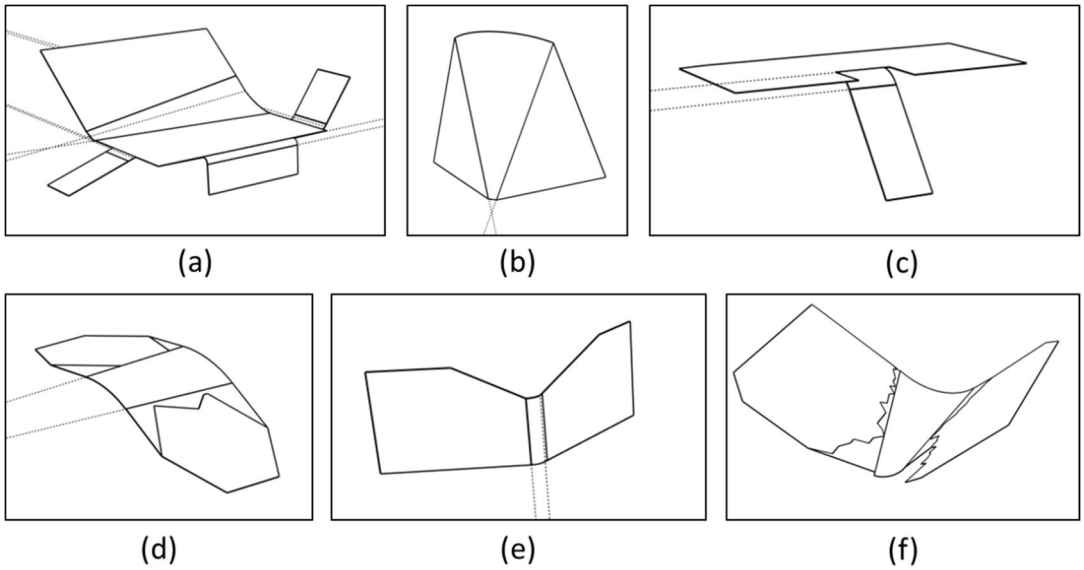


Figure 35. The resulting bends of the test cases

As can be seen in Figure 35, the test case (b) does not result in the geometry needed for a lofted bend. The reason is, that the method presented in this thesis only supports right cones. However, the cone section needed in a lofted bend needs to be non-right. Therefore to fully support lofted bends, an intuitive way to define non-right cones as well as a way to use them when attaching two elemental plates with a bent section would have to be determined.

6 CONCLUSIONS

The objective of this thesis was to introduce a modeling concept for bent steel plates and generating them, and to build a prototype for testing the usability of the concept developed. The thesis also aimed at reviewing relevant literature related to modeling bent steel plates in a software product. Mixed Integer Linear Programming is used in the proposed decision making algorithm to effectively find the most desirable bent plate structure that would satisfy most common use cases that were identified.

Unfolding and modeling of a bent steel plate has been studied quite extensively before, but no other papers were found in the literature review to have studied the automatic generation of bent steel plate structures in a BIM product. In the modeling approach chosen in this thesis, the generation of a bent steel plate structure was based on the user to input the flat areas of the bent plate. The software product should automatically calculate the best guess for the structure of the whole bent plate, still letting the user to edit the parameters of the bends later.

The proposed decision model can create satisfactory bent plate structures with most of the common use cases identified based on discussions with and documents from the Tekla business department. The decision model provides the software user with a good starting point for modeling curved metal elements in a BIM-model. The model works sufficiently fast and reliably.

However, it is not a tool that could be used as such without the possibility to make changes to the generated bent plate objects in the model afterwards. In an engineering tool all parameters need to be accessible manually so that specific and accurate values can be used in design. It does not take into account the physical properties of the generated object or consider its structural endurance.

6.1 FUTURE RESEARCH

The computation time of the MILP-problem solving the optimal bend is reasonably low, and constitutes no issues in ordinary use cases. However, the computing time of the MILP-problem for solving the minimum length bend is relatively long compared to the

calculation of the seemingly more complex optimal bend topology problem. The minimum length bend problem should be reformulated as proposed in Section 5.

The model developed in this thesis could be used in a BIM software product for generating model objects of bent metal plate structures. The decision model and modeling approach presented in this thesis is not material-dependent, so it could also be used with other materials that can be bent, such as plastics, paper and cardboard. The approach has potential for being used in other industries than building and construction too, such as ship building, packing or industrial design. Using the modeling approach for other purposes would require further research.

One useful byproduct of formulating the bend fitting problem as a linear programming model is that it could be used for validating whether the bend parameters are feasible. Because the algorithm solving linear programming problems works so that it verifies that the problem is feasible, the bend parameters could be added to the linear programming model's constraints, and solve the optimization model with some objective function (such as a constant). The optimization problem developed could be modified to solve the constraint programming problem for finding whether given parameters are feasible.

Currently the model developed is based on the judgment and materials provided by the business unit of Tekla. The algorithm developed in the thesis could be improved by conducting a more extensive research on typical use cases of bent steel plate structures. Also, there still remains a great deal to be studied in the optimal design of the user interaction of the tool creating bent plate structures.

As presented in this thesis, the proposed decision model only supports curved sections of the shape of a circular cylinder or a right circular cone. This is a restrictive limitation and there are many other possible developable surface geometries that could be used. Different developable surface parameterizations and methods the users could define them intuitively still remain to be studied. Also, in this thesis only two different parameterizations for lateral boundaries of the bend are presented. Many relevant lateral boundary parameterizations would still need to be studied.

BIBLIOGRAPHY

Alves, M. J. & Clímaco, J., 2007. A review of interactive methods for multiobjective integer and mixed-integer programming. *European Journal of Operational Research,* Volume 180, pp. 99--115.

Appa, G. M., Pitsoulis, L. S. & Williams, H. P., 2006. *Handbook on Modelling for Discrete Optimization.* New York: Springer.

Beggs, J. S., 1983. *Kinematics.* Berlin: Springer.

Bertsimas, D. & Tsitsiklis, J. N., 1997. *Introduction to Linear Optimization.* Belmont: Athena Scientific.

Bo, P. & Wang, W., 2007. Geodesic-controlled developable surfaces for modeling paper bending. *Computer Graphics Forum,* Volume 26, pp. 365--374.

Brisson, E., 1989. Representing geometric structures in d dimensions: topology and order. *Proceedings of the Fifth Annual Symposium on Computational Geometry,* pp. 218--227.

Cai, Z.-Y., Li, M.-Z. & Lan, Y.-W., 2012. Three-dimensional sheet metal continuous forming process based on flexible roll bending: Principle and experiments. *Journal of Materials Processing Technology,* Volume 212, pp. 120--127.

Camp, C., Pezeshk, S. & Cao, G., 1998. Optimized design of two-dimensional structures using a genetic algorithm. *Journal of Structural Engineering,* Volume 124, pp. 551--559.

Chuang, S. H. & Huang, S., 1996. Feature decomposition from solid models for automatic flattening. *Computer-Aided Design,* Volume 28, pp. 473--481.

Ciarlet, P. G., 2005. An introduction to differential geometry with applications to elasticity. *Journal of Elasticity,* Volume 78, pp. 1--215.

Clements, J. C. & Leon, L. J., 1987. A fast, accurate algorithm for the isometric mapping of a developable surface. *SIAM Journal on Mathematical Analysis,* Volume 18, pp. 966--971.

Costa, L. & Oliveira, P., 2001. Evolutionary algorithms approach to the solution of mixed integer non-linear programming problems. *Computers & Chemical Engineering,* Volume 25, pp. 257--266.

Eigensatz, M. et al., 2010. Paneling architectural freeform surfaces. *ACM Transactions on Graphics (TOG),* Volume 29, p. 45.

Grossmann, I. E., 2002. Review of nonlinear mixed-integer and disjunctive programming techniques. *Optimization and Engineering,* Volume 3, pp. 227--252.

Helgason, S., 1979. *Differential Geometry, Lie Groups, and Symmetric Spaces.* New York: Academic press.

Hoffmann, C. M. & Rossignac, J. R., 1996. A road map to solid modeling. *Visualization and Computer Graphics,* Volume 2, pp. 3--10.

Huang, C.-W. & Shih, T.-Y., 1997. On the complexity of point-in-polygon algorithms. *Computers & Geosciences,* Volume 23, pp. 109--118.

Kirk, D. E., 1970. *Optimal Control Theory: An Introduction.* Englewood Cliffs: Prentice-Hall.

Liu, W. & Tai, K., 2007. Optimal design of flat patterns for 3D folded structures by unfolding with topological validation. *Computer-Aided Design,* Volume 39, pp. 898--913.

Marler, R. T. & Arora, J. S., 2004. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization,* Volume 26, pp. 369--395.

Mäntylä, M., 1988. *An Introduction to Solid Modeling.* Rockville: Computer science press.

Nádai, A., 1950. *Theory of Flow and Fracture of Solids.* New York: McGraw-Hill.

O'Rourke, J., 2000. Folding and unfolding in computational geometry. *Discrete and Computational Geometry*, pp. 258--266.

Pekerman, D., Elber, G. & Kim, M.-S., 2008. Self-intersection detection and elimination in freeform curves and surfaces. *Computer-Aided Design,* Volume 40, pp. 150--159.

Peternell, M., 2004. Developable surface fitting to point clouds. *Computer Aided Geometric Design,* Volume 21, pp. 785--803.

Pottmann, H. et al., 2008. Freeform surfaces from single curved panels. *ACM Transactions on Graphics (TOG),* Volume 27, p. 76.

Press, W. H., Teukolsky, S. A., Vetterling, W. T. & Flannery, B. P., 2002. *Numerical Recipes in C++: the Art of Scientific Computing, Second Edition.* Cambridge: Cambridge University Press.

Reddy, J. N., 2006. *Theory and Analysis of Elastic Plates and Shells.* Philadelphia: Taylor & Francis.

Robbin, J. W., Madison, U. & Salamon, D. A., 2013. Introduction to Differential Geometry, Lecture Notes, ETH.

Shamos, M. I. & Hoey, D., 1976. Geometric intersection problems. *Foundations of Computer Science,* pp. 208--215.

Stroud, I., 2006. *Boundary Representation Modelling Techniques.* London: Springer.

Tai, K., Liu, W. & Thimm, G. L., 2004. Unfolding and flat layout design of non-manifold 3D folded structures. *Computer-Aided Design & Applications,* Volume 1.

The Digital University, 2012. *Analytical Mechanics Video #5: Geodesics On A Cone.* [Online]
Available at: https://www.youtube.com/watch?v=OKLqB4SQ92k
[Accessed April 2014].

Tilove, R. B., 1980. Set membership classification: A unified approach to geometric intersection problems. *Computers, IEEE Transactions on,* Volume 100, pp. 874--883.

Timoshenko, S. P. & Gere, J. M., 1961. *Theory of Elastic Stability.* New York: McGraw-Hill.

Timoshenko, S., Woinowsky-Krieger, S. & Woinowsky, S., 1959. *Theory of Plates and Shells.* New York: McGraw-Hill.

Tongxi, X. Y. & Zhang, L., 1996. *Plastic Bending: Theory and Applications.* Singapore: World Scientific.

Wang, C.-H., 1997. Manufacturability-driven decomposition of sheet metal products. Doctoral dissertation, Carnegie Mellon University.

Voelcker, H. & Requicha, A., 1977. Constructive solid geometry. Technical memo, University of Rochester.

Wolter, K. H., 1952. *Freies Biegen von Blechen: mit 4 Zahlentafeln.* Düsseldorf: Dt. Ingenieur-Verlag.

Yang, R. & Chuang, C., 1994. Optimal topology design using linear programming. *Computers & Structures,* Volume 52, pp. 265--275.

Yeh, S., Kamran, M. & Nnaji, B., 1995. Unfolding sheet metal parts: a graph-based approach. *The International Journal of Production Research,* Volume 33, pp. 729--740.

Zhang, P., Guo, B., Shan, D.-B. & Ji, Z., 2007. FE simulation of laser curve bending of sheet metals. *Journal of Materials Processing Technology,* Volume 184, pp. 157--162.

APPENDIX 1. ALGORITHM FOR FINDING OVERLAPPING SECTIONS OF A LINE AND A POLYGON

The algorithm presented here finds the intersection points of a polygon and a line, and then determines which sections of the line are within and which sections are without the polygon. This problem is known as *line/polygon clipping* (Tilove, 1980).

The idea is to calculate all the intersection points of the polygon and the line. Thereafter, the intersection points are ordered in terms of the direction vector of the line. Then the intersection points are categorized into two categories based on the (out-pointing) normals of the polygon edge: (1) the line entering the polygon area, (2) the line exiting the polygon area. Finally the type of the section between each pair of consecutive points is determined by inferring from the intersection point values.

The cases where the line is going along an edge is considered "outside" of the polygon. This is because the algorithm is used for verifying the boundaries of a curved section, and it is clear that there is nothing wrong about the curved section boundary going along a polygon edge.

The algorithm is presented in pseudo code as follows:

```
// go through edges, find intersections and type them
FOR EACH edge IN edges DO
  IF intersects(line, edge) THEN
    exiting := dot(line.direction, edge.out_pointing_normal)
    IF exiting < 0 THEN
      section_points.add(ENTER, dot(intersectionpoint, line.direction))
    ELSE IF exiting > 0 THEN
      section_points.add(EXIT, dot(intersectionpoint, line.direction))
    END IF
  END IF
LOOP
//sort intersection points according to their line coordinates and create sections
sort_by_last_column(section_points)
sections := new ArrayOfSections()
sections.add(WITHOUT)
sections.last.start_point := -infinity
FOR EACH point IN section_points DO
  sections.last.end_point := point[1]
  IF point[0] == ENTER THEN
    sections.add(WITHIN)
  ELSE
    sections.add(WITHOUT)
  END IF
  sections.last.start_point := point[1]
LOOP
sections.last.end_point := +infinity
```

APPENDIX 2. FINDING A GEODESIC OF A CONE

Here we find the geodesic of a cone using calculus of variation. First, using the approach given in and modifying slightly the tutorial by The Digital University (2012) Euler-Lagrange equation is used to find the minimum. Then the constant values derived for known start and end points of the geodesic.

In cylindrical coordinates a cone, whose axis is aligned with the z-axis can be expressed as follows

$$z^2 = m^2 R^2.$$

The following relationship holds for the variation of arc length in in a cylindrical coordinate system

$$ds^2 = dR^2 + R^2 d\theta^2 + dz^2.$$

The component dz^2 can be substituted with the term $m^2 dR^2$, and the arc length equation can be formulated to be

$$ds = \sqrt{(1 + m^2)dR^2 + R^2 d\theta^2}.$$

By dividing all the terms under the square root sign by dR^2 and integrating, arc length can be expressed as

$$\int ds = \int \sqrt{(1 + m^2) + R^2 \left(\frac{\partial \theta}{\partial R}\right)^2}\, dR.$$

A simple variation problem can be formulated as follows. Find the function x^* that minimizes or maximizes the functional

$$J(x) = \int_{t_0}^{t_1} g(x(t), \dot{x}(t), t)dt.$$

For x to be extremal, the Euler equation has to hold true (Kirk, 1970).

$$\frac{\partial g}{\partial x}(x(t), \dot{x}(t), t\,) - \frac{d}{dt}\left[\frac{\partial g}{\partial \dot{x}}(x(t), \dot{x}(t), t\,)\right] = 0.$$

As can be seen, the arc length can be interpreted as a functional of this sort and so, the above Euler equation also has to hold for the geodesic expressed as the azimuth angle $\theta(R)$. Let us denote $\frac{\partial\theta}{\partial R} = \dot{\theta}$.

$$\frac{d}{dR}\left[\frac{\partial}{\partial\dot{\theta}}\left(\sqrt{(1+m^2)+R^2\dot{\theta}^2}\right)\right] = 0$$

The partial derivate over $\dot{\theta}$ is

$$\frac{\partial}{\partial\dot{\theta}}\left(\sqrt{(1+m^2)+R^2\dot{\theta}^2}\right) = \frac{R^2\dot{\theta}}{\sqrt{(1+m^2)+R^2\dot{\theta}^2}}.$$

This has to equal to a constant. Let us denote that constant by C_2 so that

$$\left((1+m^2)+R^2\dot{\theta}^2\right)C_2^2 = R^4\dot{\theta}^2\,.$$

This can be further simplified to

$$\dot{\theta}^2 = \frac{(1+m^2)C_2^2}{R^2(R^2-C_2^2)}$$

$$\dot{\theta} = \pm\frac{\sqrt{(1+m^2)}C_2}{R\sqrt{R^2-C_2^2}}.$$

Because $\dot{\theta} = \frac{\partial\theta}{\partial R}$, the expression above becomes

$$\int d\theta = \pm\sqrt{(1+m^2)}C_2\int\frac{dR}{R\sqrt{R^2-C_2^2}}.$$

To integrate the latter integral above, let R be substituted by a secant function of a substitute variable ϕ

$$R = C_2\sec(\phi)$$
$$dR = C_2\sec(\phi)\tan(\phi)\,d\phi$$
$$\phi = \sec^{-1}\left(\frac{R}{C_2}\right) = \cos^{-1}\left(\frac{C_2}{R}\right).$$

Therefore the integration can be expressed as

$$\int d\theta = \pm\sqrt{(1+m^2)}C_2\int\frac{C_2\sec(\phi)\tan(\phi)}{C_2\sec(\phi)\,C_2\sqrt{\sec^2(\phi)-1}}d\phi\,.$$

Because $\sec^2(x) - 1 = \tan^2(x)$, the expression can be simplified and integrated

$$\int d\theta = \pm\sqrt{(1+m^2)} \int \frac{\sec(\phi)\tan(\phi)}{\sec(\phi)\tan(\phi)} d\phi$$

$$\theta + C_1 = \pm\sqrt{(1+m^2)}\phi,$$

where C_1 is the integration constant. The $\pm$ sign only defines the direction of the geodesic, which is irrelevant, so the expression can be simplified

$$\theta = \sqrt{1+m^2}\,\sec^{-1}\left(\frac{R}{C_2}\right) - C_1\,.$$

Getting rid of the inverse secant function and solving the equation for C_2, we get

$$\cos\left(\frac{\theta + C_1}{\sqrt{1+m^2}}\right) = \frac{C_2}{R}\,,$$

which is equivalent to:

$$R\cos\left(\frac{\theta + C_1}{\sqrt{1+m^2}}\right) = C_2\,.$$

This equation has to hold for both end points of the geodesic. Let the radial component of the first point of the geodesic be denoted R_1 and for the second point R_2. Because the geodesic end points are an equal angle from the intersection line of the planes, the azimuth angles are $-\theta$ and θ, i.e.,

$$R_1\cos\left(\frac{-\theta + C_1}{\sqrt{1+m^2}}\right) = R_2\cos\left(\frac{\theta + C_1}{\sqrt{1+m^2}}\right)\,.$$

Let $\frac{\theta}{\sqrt{1+m^2}}$ be denoted $\hat{\theta}$ and $\frac{C_1}{\sqrt{1+m^2}}$ be denoted $\hat{C}_1$

$$R_1\cos\left(-\hat{\theta} + \hat{C}_1\right) = R_2\cos\left(\hat{\theta} + \hat{C}_1\right).$$

We solve this for $\hat{C}_1$

$$R_1\cos -\hat{\theta}\cos\hat{C}_1 - R_1\sin -\hat{\theta}\sin\hat{C}_1 = R_2\cos\hat{\theta}\cos\hat{C}_1 - R_2\sin\hat{\theta}\sin\hat{C}_1$$

$$\tan\hat{C}_1 = \frac{\left(R_2\cos\hat{\theta} - R_1\cos\hat{\theta}\right)}{\left(R_2\sin\hat{\theta} + R_1\sin\hat{\theta}\right)}.$$

Finally C_1 can be solved based on $\hat{C}_1$

$$\tan\frac{C_1}{\sqrt{1+m^2}} = \frac{\left(R_2\cos\frac{\theta}{\sqrt{1+m^2}} - R_1\cos\frac{\theta}{\sqrt{1+m^2}}\right)}{\left(R_2\sin\frac{\theta}{\sqrt{1+m^2}} + R_1\sin\frac{\theta}{\sqrt{1+m^2}}\right)}$$

$$C_1 = \sqrt{1+m^2}\tan^{-1}\left(\frac{R_2-R_1}{R_2+R_1}\cot\frac{\theta}{\sqrt{1+m^2}}\right).$$

APPENDIX 3. DERIVING THE ANGLE BETWEEN A GEODESIC AND A GENERATOR LINE ON A CONE

$$\cos\big(\alpha(\theta_L)\big) = \frac{\boldsymbol{s'}_{gd}(\theta_L) \cdot \dot{\boldsymbol{s}}_{gl}(\theta_L)}{\left\|\boldsymbol{s'}_{gd}(\theta_L)\right\| \left\|\dot{\boldsymbol{s}}_{gl}(\theta_L)\right\|}$$

The derivative of a geodesic of a right circular cone can be expressed as

$$\boldsymbol{s'_{gd}}(\theta) = (\cos(\theta)\, R'(\theta) - R(\theta)\sin(\theta))\boldsymbol{r}_0 + (\sin(\theta)\, R'(\theta) + R(\theta)\cos(\theta))\boldsymbol{r}_0 \times \boldsymbol{h}_0 + aR'^{(\theta)}\boldsymbol{h}_0.$$

The derivative of a generator line of a ruled surface is

$$\dot{\boldsymbol{s}}_{\boldsymbol{gl}}(\theta) = \frac{\partial \boldsymbol{s}_{gl}}{\partial L} = \cos\theta\, \boldsymbol{r}_0 + \sin\theta\, \boldsymbol{r}_0 \times \boldsymbol{h}_0 + a\boldsymbol{h}_0.$$

The angle between the geodesic derivative and the generator line derivative can be solved based on their dot product and the lengths of the derivative vectors. The dot product:

$$\begin{aligned}\boldsymbol{s'}_{gd}(\theta_L) \cdot \dot{\boldsymbol{s}}_{gl}(\theta_L) &= (\cos^2(\theta_L)\, R'(\theta_L) - R(\theta_L)\cos(\theta_L)\sin(\theta_L)) \\ &+ (\sin^2(\theta_L)\, R'(\theta_L) + R(\theta_L)\cos(\theta_L)\sin(\theta_L)) + a^2 R'(\theta) \\ &= (1 + a^2)R'(\theta_L).\end{aligned}$$

The lengths of the derivative vectors are

$$\begin{aligned}\left\|\boldsymbol{s'}_{gd}(\theta_L)\right\| &= (\cos^2(\theta)\, R'^2(\theta) - 2\cos(\theta)\sin(\theta)\, R(\theta)R'(\theta) + \sin^2(\theta)\, R^2(\theta) \\ &+ \sin^2(\theta)\, R'^2(\theta) + 2\cos(\theta)\sin(\theta)\, R(\theta)R'(\theta) + \cos^2(\theta)\, R^2(\theta) \\ &+ a^2 R'^2(\theta))^{\frac{1}{2}} = \big((a^2+1)R'^2(\theta) + R^2(\theta)\big)^{\frac{1}{2}}\end{aligned}$$

$$\left\|\dot{\boldsymbol{s}}_{gl}(\theta_L)\right\| = (a^2+1)^{\frac{1}{2}}.$$

The cosine of the angle between the vectors is the dot product of the vectors divided by the product of the vector lengths

$$\cos\big(\alpha(\theta_L)\big) = \frac{(a^2+1)R'(\theta_L)}{\big((a^2+1)^2 R'^2(\theta_L) + (a^2+1)R^2(\theta_L)\big)^{\frac{1}{2}}} = \frac{R'(\theta_L)}{\left(R'^2(\theta_L) + \frac{R^2(\theta_L)}{(a^2+1)}\right)^{\frac{1}{2}}}.$$

APPENDIX 4.UNFOLDED STRAIGHT LINE BORDERS IN A CONICAL BEND

The lateral boundary coordinates on the unfolded plane are a straight line

$$y_{\max} = A + Bx_{\max}\,.$$

Getting the values of $y_{\max}$ and $x_{\max}$ from (67), the equation above can be expanded

$$-\sqrt{a^2+1}\,R(\theta_L)C = A + B(L(\theta_L) + (a^2+1)R'(\theta_L)C)$$
$$-\left(\sqrt{a^2+1}\,R(\theta_L) + (a^2+1)R'(\theta_L)B\right)C = A + BL(\theta_L),$$

where C is an auxiliary variable. The equation above is solved for C:

$$C = -\frac{A + BL(\theta_L)}{\left(\sqrt{a^2+1}\,R(\theta_L) + (a^2+1)R'(\theta_L)B\right)}.$$

The value of C is expanded

$$\frac{R_{\max}(\theta_L) - R(\theta_L)}{\sqrt{(a^2+1)R'^2(\theta_L) + R^2(\theta_L)}} = -\frac{A + BL(\theta_L)}{\left(\sqrt{a^2+1}\,R(\theta_L) + (a^2+1)R'(\theta_L)B\right)}.$$

Solving it for $R_{\max}(\theta_L)$ gives the following equation

$$R_{\max}(\theta_L) = R(\theta_L) - \left(A + BL(\theta_L)\right)\frac{\sqrt{(a^2+1)R'^2(\theta_L) + R^2(\theta_L)}}{\sqrt{a^2+1}\,R(\theta_L) + (a^2+1)R'(\theta_L)B}.$$

The value of $R(\theta)$ and its derivative, its squared form and the squared form derivative are

$$R(\theta) = \frac{C_2}{\cos\left(\frac{\theta+C_1}{\sqrt{1+a}}\right)},\ R^2 = \frac{C_2^2}{\cos^2\left(\frac{\theta+C_1}{\sqrt{1+a}}\right)},\ R'(\theta) = \frac{C_2}{\sqrt{1+a}}\frac{\tan\left(\frac{\theta+C_1}{\sqrt{1+a}}\right)}{\cos\left(\frac{\theta+C_1}{\sqrt{1+a}}\right)},\ R'^2(\theta) = \frac{C_2^2}{1+a}\frac{\tan^2\left(\frac{\theta+C_1}{\sqrt{1+a}}\right)}{\cos^2\left(\frac{\theta+C_1}{\sqrt{1+a}}\right)}.$$

Solving the equation gives the radius at the lateral boundary

$$R_{\max}(\theta_L) = \frac{C_2}{\cos\left(\frac{\theta_L + C_1}{\sqrt{1+a}}\right)} - \frac{A + BL(\theta_L)}{\left(\sqrt{a^2+1}\right)\cos\left(\frac{\theta_L + C_1}{\sqrt{1+a}}\right) + \frac{(a^2+1)}{\sqrt{1+a}}\sin\left(\frac{\theta_L + C_1}{\sqrt{1+a}}\right)B}.$$